(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 December 2002 (12.12.2002)

PCT

(10) International Publication Number
WO 02/099642 A1

(51) International Patent Classification[7]: G06F 11/14

(21) International Application Number: PCT/US02/07154

(22) International Filing Date: 6 March 2002 (06.03.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/291,767	17 May 2001 (17.05.2001)	US
09/862,898	21 May 2001 (21.05.2001)	US
Not furnished	22 May 2001 (22.05.2001)	US
10/075,136	19 November 2001 (19.11.2001)	US
Not furnished	11 February 2002 (11.02.2002)	US
10/090,480	27 February 2002 (27.02.2002)	US
Not furnished	27 February 2002 (27.02.2002)	US

(71) Applicant *(for all designated States except US)*: **SELF REPAIRING COMPUTERS, INC.** [US/US]; 2460 - 21st Avenue, San Francisco, CA 94116 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **LARGMAN, Kenneth** [US/US]; 2460 - 21st Avenue, San Francisco, CA 94116 (US). **MORE, Anthony, B.** [US/US]; 750 Warfield AVenue, #504, Oakland, CA 94610 (US). **BLAIR, Jeffrey** [US/US]; #6 El Sereno Court, San Francisco, CA 94116 (US).

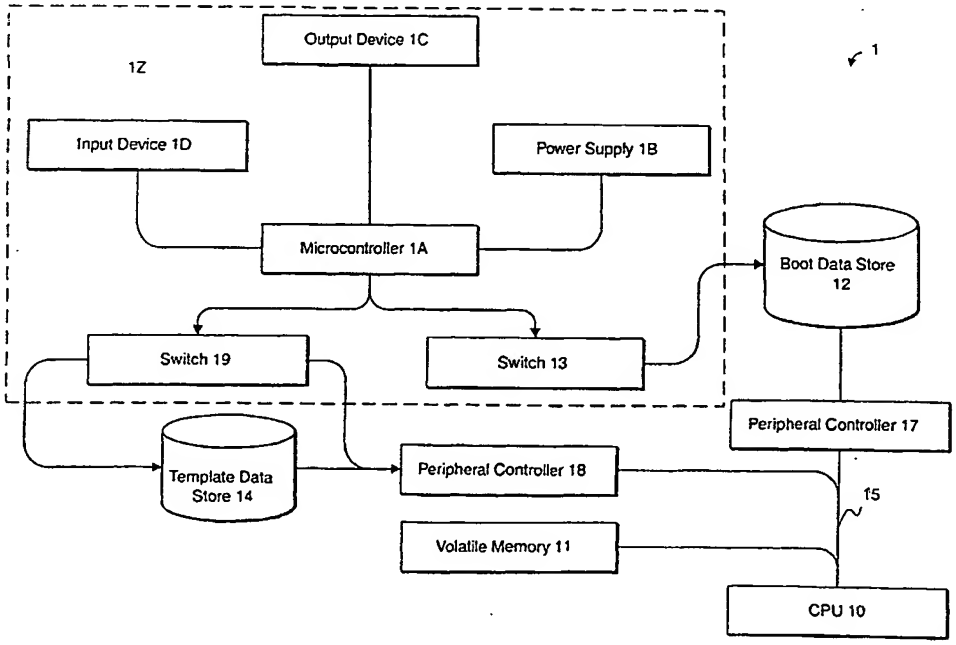
(74) Agents: **ANANIAN, Michael, R.** et al.; Flehr Hohback Test Albritton & Herbert LLP, 4 Embarcadero Center, Suite 3400, San Francisco, CA 94111-4187 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: A COMPUTER WITH FAULT-TOLERANT BOOTING



(57) Abstract: A method for a computer repairing itself, the method comprising the computer-executed steps of: booting from a first boot device; then, in response to a signal indicating a need for repair, booting from a second boot device; and then repairing software on the first boot device while booted from the second boot device.

WO 02/099642 A1

Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A COMPUTER WITH FAULT-TOLERANT BOOTING

This invention relates to computers, computer repair and computer architecture. More particularly, the invention relates to a computer architecture and software that enables the computer to repair itself.

BENEFIT APPLICATIONS

This application claims the benefit of the following applications:

U.S. Provisional Patent Application No. 60/291767, entitled, "A Self-Repairing Computer," filed May 17, 2001, naming Kenneth Largman and Anthony B. More as inventors, with Attorney Docket No. P-70543-2/RMA/LM, and commonly assigned to Self Repairing Computers, Inc., San Francisco, California.

U.S. Patent Application No. 09/862,898, entitled, "A Computer with Switchable Components," filed May 21, 2001, naming Kenneth Largman and Anthony B. More as inventors, with Attorney Docket No. A-70543/RMA/LM, and commonly assigned to Self Repairing Computers, Inc., San Francisco, California.

U.S. Patent Application No. (unknown), entitled, "On-The-Fly Repair Of A Computer," filed November 19, 2001, naming Kenneth Largman and Anthony B. More and Jeffrey Blair as inventors, with Attorney Docket No. A-70543-1/RMA/LM, and commonly assigned to Self Repairing Computers, Inc., San Francisco, California.

U.S. Patent Application No. (unknown), entitled, "External Repair Of A Computer," filed February 11, 2002, naming Kenneth Largman

and Anthony B. More and Jeffrey Blair as inventors, with Attorney Docket No. A-70543-2/RMA/LM, and commonly assigned to Self Repairing Computers, Inc., San Francisco, California.

U.S. Patent Application No. (unknown), entitled, "Backup Of A Computer," filed February 27, 2002, naming Kenneth Largman and Anthony B. More and Jeffrey Blair as inventors, with Attorney Docket No. A-70543-3/RMA/LM, and commonly assigned to Self Repairing Computers, Inc., San Francisco, California.

International Application No. (unknown), entitled, "A Computer With Switchable Components," filed February 27, 2002, naming Kenneth Largman and Anthony B. More and Jeffrey Blair as inventors, with Attorney Docket No. FP-70543-001/RMA/LM, and commonly assigned to Self Repairing Computers, Inc., San Francisco, California.

International Application No. PCT/ US 01/ 16629, entitled, "A Computer With Switchable Components," filed May 22, 2001, naming Kenneth Largman and Anthony B. More and Jeffrey Blair as inventors, with Attorney Docket No. FP-70543-PC/RMA/LM, and commonly assigned to Self Repairing Computers, Inc., San Francisco, California.

U.S. Patent Applications Nos. 60/291767, 09/862,898, (unknown) filed November 19, 2001, (unknown) filed on February 11, 2002, (unknown) filed February 27, 2002, International Application No. (unknown) filed February 27, 2002, and International Application No. PCT/ US 01/ 16629 are incorporated by reference herein.

BACKGROUND

Personal-computer manufacturers and sellers often offer via-telephone and on-site repair services. Yet purchasers — particularly home, home-office and small-office purchasers — readily complain that their service contract offers less service than they expected. For example, a

computer seller may dispatch a technician only after the purchaser calls the help center, performs a number of tests under the direction of the help center, escalates the problem at the telephone help center and performs redundant or additional tests under the direction of a putatively more knowledgeable telephone-help staff. The purchaser may have to escalate the problem still further and perform additional redundant tests before a repair technician is dispatched.

Frequently, the help center directs the customer to cycle the power on the computer, to re-boot the computer, to detach and reattach peripherals in question and to re-install application and operating-system software. Each call to the help center and each level of escalation may require the purchaser to cycle, re-boot, detach and reattach.

Detaching and reattaching peripherals can be extremely inconvenient. USB devices, for example, typically attach at the back of a computer in a location difficult to reach. In any event, the non-digerati purchaser may fear disassembling his computer, worrying that he may damage the computer further.

Help centers even direct a customer to reformat the boot drive of the computer and re-install operating-system and application software. Re-formatting is an onerous task for several reasons. Firstly, the home, home-office and small-office user rarely reformats a drive in the normal operation of his computer and is unfamiliar with the process itself. Secondly, reformatting destroys all the data on the drive, and such a user understandably becomes anxious on finding out that he will lose all of his data. Thirdly, such a user may not retain the application or operating-system installation media, especially where the seller pre-installs the software. The user may have been unsure which media to keep, or intending to keep a particular media, is in fact unable to locate that media later when needed.

Fourthly, the user typically does not back up his drives as often

as an information technologist would recommend. That he will have to rely on his back ups (if any) if he is to have any hope of restoring his application is then not a comforting thought.

Accordingly, the art evinces a need for a computer that reduces or even eliminates the need for a user to call a help line, to keep installation media, to attach and reattach peripherals at the port, etc. Indeed, a computer that reduces or eliminates the technical savvy its user needs to effect repairs is desirable.

These and other goals of the invention will be readily apparent to one of ordinary skill in the art on reading the background above and the description below.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a computer incorporating an embodiment of the invention.

Figure 2 is a schematic of a data-store switch according to an embodiment of the invention.

Figures 3A through **3B** illustrate the switch-and-repair process according to one embodiment of the invention.

Figure 4 illustrates the flow of control in a data-store switch according to one embodiment of the invention.

Figure 5 illustrates a computer incorporating an embodiment of the invention.

Figures 6A, 6B illustrate a computer incorporating an embodiment of the invention. **Figure 6A** illustrates the enabling of a data store in conjunction with the defeat of access to a communications link. **Figure 6B** illustrates the enabling of a data store in order to support access to the communications link.

Figures 7A, 7B illustrate a computer incorporating an

embodiment of the invention. **Figure 7A** illustrates the computer in its Network Disconnected state, while **Figure 7B** illustrates the computer in its Network Connected state.

Figure 8 illustrates a computer incorporating an embodiment of the invention.

Figures 9A, 9B illustrate a computer incorporating embodiments of the invention.

Figure 10 illustrates a computer incorporating an embodiment of the invention.

(The drawings are not to scale.)

SUMMARY

Herein are taught apparatus and methods for a computer to repair itself.

DESCRIPTION OF THE INVENTION

OVERVIEW

An example of the invention in use follows: A user runs an application on a computer incorporating an embodiment of the invention. At some point, the user modifies the application or underlying operating system to the point that the application, the operating system or both become unusable. Indeed, the user may no longer be able to even boot the operating system.

Recognizing that the computer needs to be repaired, the user throws a switch on the computer. The computer fixes the malfunctioning software and so informs the user.

The user can then re-boot the computer. On re-booting, the user again has access to a correctly functioning operating system, application and data files.

A SELF-REPAIRING COMPUTER

Figure 1 illustrates a computer **1** incorporating an embodiment of the invention. The computer **1** may include a CPU **10**, volatile memory **11**, peripheral controllers **17, 18**, a first non-volatile data store **12** and a bus **15**, all well known in the art.

The computer **1** may also include switches **13, 19**, a second non-volatile data store **14**, a controller **1A**, a power supply **1B**, an output device **1C** and an input device **1D**.

The bus **15** may communicatively couple the volatile memory **11** and the peripheral controllers **17, 18** to each other and to the CPU **10**. The peripheral controllers **17, 18** may communicatively couple with the data stores **12, 14**, respectively.

The switches **13, 19**, the controller **1A**, power supply **1B**, output device **1C** and input device **1D** may form a data-store switch **1Z**. A data-store switch may alter the accessibility of a connected data store according to the setting of the switch.

The controller **1A** may communicatively couple with the switches **13, 19**, the output device **1C** and the input device **1D**. The power supply **1B** may supply the controller **1A** (and other switch components) with power. More particularly, the power supply **1B** may power the controller **1A** independently of the power to the rest of the computer **1**.

The power to the switch **1Z** may come from the same source as the power for the rest of the computer (the wall outlet or laptop battery, for example). The switch **1Z** may then be powered from that supply even when the rest of the computer **1** is not. **Figure 10** illustrates this embodiment of the invention.

The switch **13** may communicate with the data store **12**. The switch may control (toggle, for example) the identification settings of the data store **12**.

The switch **19** may couple to the data store **14**. The switch **19** may control (toggle, for example) the power to the data store **14**.

The volatile memory **11** may be random-access memory. The data stores **12, 14** may be magnetic disks, for example.

The output device **1C** may be the monitor of the computer **1**, LEDs or an LCD distinct from the monitor, for example.

Figure 2 is a schematic of the data-store switch **1Z** according to an embodiment of the invention. In **Figure 2**, the opto-isolators **U2, U3** implement the switches **13, 19**, respectively. The Basic Stamp II microcontroller **U1** (from Parallax, Inc., Rocklin, California) implements the controller **1A**. The battery **V3** implements the power supply **1B**. The LCD display port **J1** represents the output device **1C**, and the switches **S1, S2** implement the input device **1D**. (Opto-isolator **U4** detects whether the computer **1** has power.)

In a first mode of operation herein termed "normal mode," the computer **1** may run a predetermined operating system and application. Accordingly, the data store **12** may contain a correctly functioning copy of that software. The CPU **10** may access the data store **12**, boot the operating system and then execute that application.

The data store **12** is termed herein the "boot data store." The data store **12** may contain a bootable, executable operating system and executable application.

The data-store switch **1Z** may make the data store **12** accessible to the computer **1** as the boot drive (by means of the switch **13**, for example). The data-store switch **1Z** may also make the data store **14** inaccessible to the computer **1** (by means of the switch **19**, for example). Otherwise, the data-store switch **1Z** may idle, waiting for user input on the device **1D**.

In the normal stage, the computer **1** may perform as a conventional computer. The user may run his application software,

inattentive to the invention incorporated into the computer **1**.

In a third mode of operation herein termed the "repair mode," the CPU **10** may run software on the data store **14** and the controller **1A** may execute a program in parallel. A mode intermediate to the normal and repair modes, herein termed the "switching mode," may effect the transition from normal to repair mode.

In the switching mode, using an input device such as the device **1D** the user may indicate that he wishes to repair software on the data store **12**. (**Figures 3A** and **3B** illustrate the switch-and-repair process according to one embodiment of the invention.) In response to the input, the computer **1** may switch from normal operation to repair, step **310**, and repair the software on the data store **12**, step **320**.

The switching of a data store may be logical or physical. Logical switching is switching enforced purely by software. For example, software may set one or more predetermined bits that it or other software tests to determine whether a data store is accessible at any given time.

A physical switch opens or closes a predetermined electrical circuit of a device to be switched. A physical switch may, for example, alter the open/close state of identification jumpers of a data store. A physical switch may turn on or off the power supply to a device to be switched.

Figure 4 illustrates the flow of control in a data-store switch **1Z** according to one embodiment of the invention. On start up, the data-store switch **1Z** may go into normal mode of operation. In this stage, the switch **1Z** may set the switch **13** to make the data store **12** the boot drive, step **4A3**. The switch also may set the switch **19** to leave the template data store **14** unpowered.

The data-store switch **1Z** may then idle, waiting for the user to initiate the switch to repair mode, step **4A5**. The data-store switch **1Z** may display a message indicating that it is in normal mode, step **4A1**.

When the data-store switch **1Z** receives an indication to switch to repair mode, the switch **1Z** may ask the user to confirm this indication, step **4B5**. Confirmation is preferable where the repair process is destructive before it is constructive. Confirmation is preferable also because the activation of the input device indicating the switch to repair mode may have been accidental or ill considered.

On confirmation if requested, the data-store switch **1Z** may switch power to the data store **14**, step **4B9**, making the data store **14** accessible to the computer **1**. The data store **14** may be permanently configured to be addressable as the boot drive when it is accessible. Accordingly, the address of the data store **12** may then change.

In normal operation, the data store **12** may be addressable as the boot drive. However, during the switch, the switch **1Z** may change the identity (address jumpers, for example) of the data store **12** to something other than the boot-drive identity.

The computer **1** is now ready to enter the repair stage.

Switched physically to repair mode, the computer **1** may boot from the template boot drive. The booted program or some other program executed during the boot sequence (autoexec.bat, for example, on machines running Windows™ operating system from Microsoft Corp., Redmond, WA) may query the user.

In one embodiment, on rebooting the computer **1** may automatically repair the data drive **12**. It copies software from the template data store **14** to the data store **12** without further direction from the user. Previously set user preferences may, however, direct the course of repair.

Thus, where the template data store **14** contains only application software, the repair process may copy over or re-install that application software from the template data store **12**. Where the template data store contains operating-system and application software, the repair

process may copy over or re-install the operating system first and then the application software.

Uninstallation or deletion of an application may precede re-installation or copying over of that software. Re-formatting of the data store **12** may precede re-installation or copying over of the operating system. Resetting of ROM-resident parameters may precede re-installation or copying over of operating-system or application software.

On completion of the repair, the repair software may direct the user to switch back to normal mode and re-boot the computer **1**.

Alternatively, the repair process may be menu-driven. The repair process may present the user a sequence of options to determine what repair process to execute. For example, on re-boot in repair mode, the repair software may offer the choices of running the repair process, reviewing repair-process settings, updating the template software (the application, operating system or repair-process software itself) and quitting the repair process.

The template data store **14** may contain application software, operating-system software and repair-process software. The application software may include the executable software itself (.exe, .dll, .o, etc.) or the files created by the application (.wpd files for Corel WordPerfect word-processing software, for example).

The software on a template data store **14** typically is an operating system and may include one or more applications, along with the underlying software to run the operating system (and any included application) on a computer with a predetermined configuration. The underlying software may include one or more boot records, one or more partition tables or a BIOS.

The template software is created by installing software onto a data store, by copying installed software onto the data store or by copying installation software onto a data store. (Installed software includes data

files and other pre-existing software.)

The template data store software may be updated. Where the template software is installation-ready software, that installation software may be updated to a different, usually later, version. Where the template software is a backup of the software on the data store **12**, a different, usually more recent, backup of the data-store software replaces or supplements that software.

Repair-process settings may include whether to recover data, run a virus check, reformat the data store, revert to a backup, run a human-mediated (i.e., manual) or an automatic repair, run diagnostics (software or hardware, for example). Repair-process settings may also include whether to format and at what level (quick versus low-level, for example), what software to re-install (operating system (OS) only; OS and executable-application software; OS, executable-application software and application data files; data files only, for example), whether to switch automatically (i.e., under program or hardware control), what level of repair to run (quick, better or best, in one embodiment), whence to setup (backup or template, in one embodiment) and whence to recover data files (most recent backup prior to repair, backup at the time of repair, other predetermined backup, query-and-response-specified backup, as examples).

The repair process may entail recovering a usable version of the appropriate data file. In some instances of computer repair, the problem is not so much with the operating-system or executable-application software so much as with the files (usually data files) associated with one or more of the applications. If the application in question is Microsoft Outlook, then the file to be recovered may be the mail-and-folder-data .pst file. Where the application is Microsoft's Internet Explorer, the file to recover may be the favorites file.

Running a virus check may entail first checking that the virus-

check-and-repair software is up to date. Because new software attacks appear daily, and because newer malicious code has a higher chance of delivering a payload, this is not a trivial step. The software may then check for malicious code and repair software, as directed by the user or by default.

The above process presupposes that the data store **14** contains a copy of (a version of) the operating-system, application software or data file on the data store **12**. In this sense, this second data store **14** is termed herein the "template data store." With the computer **1** switched to boot from the template data store **14**, the computer **1** may perform the original copying of template software onto the data store **14**. (Where the data store **14** is a read-only medium, it may arrive at the computer **1** in a pre-written state.)

An example of the operation of the computer **10** follows: Assume that the data store **12** contains a bootable Windows™ operating system (from Microsoft Corp., Redmond, WA). Assume also that the data store **12** also contains NaturallySpeaking® application software (Lernout & Hauspie, Ieper, Belgium and Burlington, MA).

The operating system and the application on the data store **12** may have each been run any number of times, and the user may have customized the operating system, the application or both to his preferences. In contrast, the template data store **14** may contain as-installed copies of the operating-system and the application software.

In the course of using his computer **1**, the user puts the computer **1** into an undesirable state. He may, for example, foul up the optional settings of the operating system or application such that he cannot reset them to a usable state. He may download a virus, trojan horse or other malicious code that changes his operating system, application or both. The particulars of the malicious code are unknown but the manifest effect is that the computer **1** is partially or completely

inoperable. He may remove files critical to the correct operation of the software. As one of skill in the art will recognize, the ways in which software may be intentionally or unintentionally altered to the point of unusability are legion.

Recognizing that his computer **1** is in an undesirable state, the user activates the switch **13**, step **300**. **Figure 3** illustrates the switch-and-repair process according to one embodiment of the invention, and step **310** illustrates the actual switching. In response to the switch activation, step **300**, the computer **1** repairs the software on the data store, step **320**.

The repair process involves copying software from the template data store **14** to the data store **14**. The software on the template data store **14** may be a master copy, a backup copy or an archive copy of software on the data store **12**. (An archive is a copy of software, which copy cannot be overwritten or deleted.)

With template software on the template data store **14**, the computer **1** may re-install or copy over software onto the data store **12**. The computer **1** may overwrite all or part of any software on the data store **12**.

The computer **1** may offer the user options as to how thorough its attempt to repair itself should be. In one embodiment, the computer **1** offers the options of a "Quick Repair," a "Better Repair," a "Best Repair" and a "Test." A Quick Repair may, for example, re-install or copy template software from the data store **14** onto the data store **12** without first re-formatting the data store **12**. The Better Repair may perform a high-level re-format of the data store **12** before that copy or re-installation. A Best Repair may perform a low-level re-format of the data store **12** before copying over or re-installing software.

Figure 4 illustrates the switch-and-repair process in more detail, according to one embodiment of the invention. The switching copies software from the template data store onto the data store, replacing the

unusable software on the data store.

A number of situations occur where the computer **1** may effect repair without rebooting. For example, if only data files or application executables need to be repaired, then shutting down the operating system booted from the data store **12** is not usually necessary — especially in newer operating systems such as Windows 2000 (Microsoft) and more sophisticated operating systems such as Linux .

Further, a large number of operating-system files can be repaired (for example, by replacement) without shutting down the operating system. Repairing the operating system without rebooting is a preferred embodiment.

Still further, for backups (automated or otherwise), continuing to run from the data store already booted may be preferable. Where the computer **1** can become sufficiently quiescent that a backup from the data store **12** to the data store **14** can occur while still booted from the data store **12**, then such a backup is quicker than shutting down and backing up the data store **12** while booted from the data store **14.**

Where the data store **12** remains the boot drive when the data store **14** is simultaneously available, the data store **14** may be addressable as other than the boot drive. The address of the data store **14** may be switched similarly to the address switching of the data store **12.**

A VIRUS- AND HACKER-RESISTANT COMPUTER

Figure 6A illustrates a computer **6** incorporating an embodiment of the invention. The computer **6** may include a CPU **60**, volatile memory **61**, peripheral controllers **67, 68**, first and second non-volatile data stores **62, 64**, data port **69**, communications link **6A** and buses **65, 66**, all well known in the art. The computer **6** may also include a data-store switch **6Z.**

The bus **65** may communicatively couple the volatile memory

61, the peripheral controllers **67, 68** and the data port **69** to each other and to the CPU **60**. The peripheral controllers **67, 68** may communicatively couple with the data stores **62, 64**, respectively. The data port **69** may mediate access to the communications link **6A**.

The bus **66** may communicatively and electrically couple the peripheral controller **67** to the data store **62** and to the boot-store switch **6Z**. More specifically, the boot-store switch **6Z** may switch the power line **661** of the bus **66**, thus powering up or down the boot store **62**.

Likewise, the bus **67** may communicatively and electrically couple the peripheral controller **68** to the data store **64** and to the boot-store switch **6Z**. The boot-store switch **6Z** may switch the power line **671** of the bus **66**, powering up or down the boot store **64**.

The port **69** may link the computer **6** to other devices such as a modems, networks, etc. as indicated by the communications link **6A**.

The computer **6** may operate in two states: Connected and Disconnected. In the Disconnected state, the computer **6** does not use the data port **69** to communicate and the data-store switch may enable the data store **62**.

By contrast, in the Connected state, the computer **6** may use the data port **69** to obtain data over the communications link **6A**. In the Connected state, the switch may enable the second data store **64**.

Thus, the computer **6** may enable only one of the multiple data stores **62, 64** at any given time, which depending on whether it is accessing the communications link **6A**. This isolates data received over the communications link **6A** to one of the data stores, namely, the data store **64**. Where the data received was maliciously created (a virus or a hacking executable), this data is confined to the data store **64**.

The switching of the data stores **62, 64** may be done under manual, hardware or software control. A mechanical throw switched by the user when the user wishes to access (or cease accessing) the

communications link exemplifies a manual switch. A boot-store switch **6Z** that responds programmatically to the CPU **60** illustrates a software-controlled switch.

For example, if the user boots an Internet browser and the communications link **6A** is the Internet, then the CPU **60** may programmatically recognize the (intended) launch of a browser and initiate the switch of the data stores **62, 64**. The switch may involve re-booting the computer **6** in order to make the second data store **64** the only data store available during the use of the communications link **6A**. (A browser on the data store **64** may launch automatically on the boot from the data store **64**.)

In one embodiment, the computer may synchronously switch the port **69** and the second boot store **64**. This may improve the resistance of the computer **6** to hacking or infection.

Figure 6A illustrates the enabling of the data store **62** in conjunction with the defeat of access to the communications link **6A**. The solid line continuing the power line **661** through the boot-store switch **6Z** illustrates the accessibility of the data store **62**. Conversely, the dashed lined through the switch **6Z** illustrates the inaccessibility of the data store **64**.

Figure 6B illustrates the enabling of the data store **64** in order to support access to the communications link **6A**. The solid power line through the boot-store switch **6Z** illustrates the accessibility of the data store **64**. Conversely, the dashed lined through the switch **6Z** illustrates the inaccessibility of the data store **62**.

The data store **64** may contain application software to process the data received over the link **6A**. In such a setting the need to migrate the data on the data store **64** to the data store **62** may be minimal or non-existent.

Where, however, the application to process the data received over the link **6A** and stored on the store **64** resides on the data store **62**,

then a process of migration is necessary. A predetermined time after receiving data over the link **6A**, the computer may simultaneously enable the data stores **62, 64** and copy the data received to the data store **62** for processing there. The delay allows, for example, anti-virus software providers to produce and distribute security software addressing threats that have come to light since the time of receipt of the data.

The migration process may be manual or automatic.

A LOCKABLE NETWORK COMPUTER

Figure 7A illustrates a computer **7** incorporating an embodiment of the invention. The computer **7** may include a CPU **70**, volatile memory **71**, a peripheral controller **77**, a non-volatile data store **72**, a data port **79**, a communications link **7A** and buses **75, 77**, all well known in the art. The computer **7** may also include a switch **7Z**.

The bus **75** may communicatively couple the volatile memory **71**, the peripheral controller **77** and the data port **79** to each other and to the CPU **70**. The peripheral controller **77** may communicatively couple with the data store **72**. The data port **79** may mediate access to the communications link **7A**.

The bus **77** may communicatively or electrically couple the data port **79** to the communications device **7B**.

The port **79** may link the computer **7** to other communicators through a communication device **7B** and over a communications link **7A**. Examples of the communications device **7B** and link **7A** include an acoustic modem **7B** and a POTS telephone line **7A**; a tap **7B** and an ethernet **7A**; and a wireless modem **7B** and radiation-permeable space **7A**.

The switch **7Z** may switch a power line **771** of the bus **77**, thus powering up or down the communications device **7B**. The switch **7Z** may switch (tri-state, for example) a data line **771** of the bus **77**, thus interrupting or enabling the ability of the communications device **7B** to transfer data to

the data port **79**.

The computer **7** may operate in two states: Network Connected and Network Disconnected. **Figure 7A** illustrates the computer **7** in its Network Disconnected state, while **Figure 7B** illustrates the computer **7** in its Network Connected state. (The solid line continuing the power line **761** through the switch **7Z** illustrates the continuity of the power or data line **771**, and dashed lined through the switch **7Z** illustrates the discontinuity of that line **771**.

In the Network Disconnected state, the switch **7Z** may disconnect the communications device **7B** from communicating on the data port **79**. Accordingly, none of the software running on the computer **7** may access the communications link **7A**.

By contrast, in the Network Connected state, the switch **7Z** may enable the communications device **7B** to communicate on the data port **79**. Accordingly, software on the computer **7** may access the communications link **7A**.

An exemplary use for the computer **7** is where a parent uses the computer **7** to access, say, his employer's computer network via a virtual private network (VPN) over the Internet **7A**. The parent also wants his child to be able to use the computer **7** for school or recreation — but without access to the Internet **7A**. The parent thus switches the computer **7** into the Network Enabled state when he (the parent) wants to use it, and switches the computer **7** into the Network Disconnected state when the child is to use the computer **7**.

The switching of the data stores **72, 74** may be done under manual, hardware or software control. A mechanical switch thrown by the user when the user wishes to access (or cease accessing) the communications link **7A** exemplifies a manual switch. A mechanical switch that may be locked with a key, for example, is preferable.

A switch **7Z** that responds programmatically to the CPU **70**

illustrates a software-controlled switch **7Z**. (The CPU **70** may respond to any kind of input, including keystrokes, voice commands, biometric data and data received over a network.) A hardware switch **7Z** may be considered as an analog computer.

A computer **7** running an operating system that supports hot swapping offers an advantage. The addition and removal of the communications device **7B** from the computer **7** may confuse OSs that do not permit hot swapping of peripherals.

A MULTI-DATA STORE SERVER

Figure 8 illustrates a computer **8** incorporating an embodiment of the invention. The computer **8** may include a CPU **80**, volatile memory **81**, a peripheral controller **87**, multiple non-volatile data stores **82a, 82b, . . . 82α**, a data port **89**, a communications link **8A** and a bus **85**, all well known in the art. The computer **8** may also include a data-store switch **8Z** and a bus **86** consisting of the buses **861** or **862**.

The bus **85** may communicatively couple the volatile memory **81**, the peripheral controller **87** and the data port **89** to each other and to the CPU **80**. The data port **89** may mediate access to the communications link **8A**.

The peripheral controller **87** may communicatively couple with the data-store switch **8Z**. The data-store switch **8Z** in turn may communicatively or electrically couple to the data stores **82**. The bus **861** may communicatively couple the data path of the switch **8Z** to those of the data stores **82**, and the bus **862** may electrically couple a power supply in or through the switch **8Z** to the data stores **82**.

The data port **89** may mediate access to the communications link **6A**. The port **89** links the computer **8** to other communicators over the communications link **7A**.

The computer **8** may operate in any of N states, where N is the number of data stores **82**. In a first state, the data-store switch **8Z** enables the first data store **82a** to communicate with the peripheral controller **87**. In the second state, the switch **8Z** enables the second data store **82b** to communicate with the peripheral controller **87**, and in the Nth state, the switch **8Z** enables the Nth data store **82α** to communicate with the peripheral controller **87**.

The corruption or other failure of the data store **82** currently communicating with the controller **87** prompts the switching from one state to another, and thus from the failed data store to another, working data store **82**. (The failed data store **82** may then be repaired in place, or it may be removed and repaired, removed and replaced, or removed permanently.)

Where, for example, the computer **9** is a web server and the communications link **8A** is the Internet, the multiple data stores **82** may provide resistance against infection and hacking by malicious users of the Internet **8A**. If the hackers succeed in corrupting the data store currently attached to the peripheral controller, then a switching may occur from that corrupted data store **82** to another correct data store **82**. This switching may occur very quickly (preferably as quickly as possible) in order to minimize the loss of access to the data on the data stores **82**.

The switching may be manual, hardware or programmatic. For example, a diagnosis program may execute periodically to determine the health of the currently accessible data store **82**.

A COMPUTER WITH PERIPHERALS THAT CAN BE CYCLED

Figure 9A illustrates a computer **9** incorporating an embodiment of the invention. The computer **9** may include a CPU **90**, volatile memory **91**, a controllers **97, 98**, a non-volatile data store **92**, a port **99**, a peripheral **9B** and buses **95, 97**, all well known in the art. The

computer **9** may also include a switch **9Z**.

The bus **95** may communicatively couple the volatile memory **91**, the controllers **97, 98** to each other and to the CPU **90**. The controller **97** may communicate with the data store **92**. The controller **98** may communicate with the peripheral **9B**.

The bus **97** may communicatively or electrically couple the port **99** (and thus the controller **98**) to the peripheral **9B**.

The peripheral **9B** may be any computer peripheral. Examples include printers, USB devices, scanners, fax machines, data stores and keyboards.

The switch **9Z** may switch a power line **971** of the bus **97**, thus powering up or down the peripheral **9B**. The switch **9Z** may switch one or more data lines **972** of the bus **97**, thus disabling or enabling the peripheral **9B** to transfer data to the port **99**.

A user of the computer **9** may be using the peripheral **9B**, transmitting or receiving data on the from the device **9B** as expected. The switch **9Z** is supplying power to the peripheral **9B**.

At some point, the computer **9** becomes unable to communicate with the peripheral **9B**. This may be caused by an error in the software or hardware of the computer **9**, including software or logic of the peripheral **9B**.

The user attempts to revive communications with the peripheral **9B**. The user may. for example, cycle the power to the peripheral **9B**. Thus, the user changes the state of the switch **9Z** such that the switch **9Z** goes from powering to the peripheral **9B**, to not powering that peripheral **9B**, to again powering that peripheral **9B**. This switching may be done manually, in hardware, or programmatically.

The cycling of the peripheral **9B** may resolve the communication problem that the user was experiencing. For example, where the problem was with the software or logic of the peripheral **9B**,

then the power cycling may clear the software or logic state of the peripheral **9B**. Where the problem was with the software or logic of the computer **1**, cycling the power may clear the software or logic state f the controller **97** or applications running in the memory **91**.

Figure 9B illustrates an alternate embodiment of the computer **9**. The switch **9Z** switches both power and data lines.

A MULTI-USER COMPUTER

Figure 5 illustrates a computer **5** incorporating an embodiment of the invention. The computer **5** may include a CPU **50**, volatile memory **51**, a peripheral controller **57**, multiple non-volatile data stores **52a, 52b, . . . 52α** and a bus **55**, all well known in the art. The computer **5** may also include a data-store switch **5Z** and a bus **56** consisting of the buses **561** or **562**.

The bus **55** may communicatively couple the volatile memory **51**, the peripheral controller **57** and the data port **59** to each other and to the CPU **50**.

The peripheral controller **57** may communicative with the data-store switch **5Z**. The data-store switch **5Z** in turn may communicatively or electrically couple with the data stores **52**. The bus **561** may communicatively couple the data path of the switch **5Z** to those of the data stores **52**, and the bus **562** may electrically couple a power supply in or through the switch**5Z** to the data stores **52**.

The computer **5** may operate in any of N states, where N is the number of data stores **52**. In a first state, the data-store switch **5Z** enables the first data store **52a** to communicate with the peripheral controller **57**. In the second state, the switch **5Z** enables the second data store **52b** to communicate with the peripheral controller **57**, and in the Nth state, the switch **5Z** enables the Nth data store **52α** to communicate with the peripheral controller **57**. Only one data store **52** may access the peripheral

controller **57** at any given time.

In one embodiment, the computer **5** has only one controller with multiple devices. In another embodiment, the computer **5'** has multiple controllers, each with respective multiple peripherals. The switching then switches among the multiple peripherals of the first controller, the multiple peripherals of the second controller, etc. (The multiple controllers need not have the same number of multiple peripherals.)

Each data store **52** may contain self-contained software for a respective user or group of users. Each data store **52** may contain a bootable operating system, and optionally such application or data files as the user(s) corresponding to the data store **52** may require or desire.

Each user or group of users may use only a predetermined one (or more) of the data stores **52**. Thus, before using the computer **5**, a user sets the switch **5Z** to the predetermined position enabling the data store **52** corresponding to that user to communicate via the controller **57**.

In this way, a first user's data is separated from a second user's data on the same computer. The computer **5** more effectively separates users' data by enforcing security at a physical level rather than at the logical (software-enforced) level typical of multi-user operating systems.

In this scenario, re-booting between switches is desirable. Re-booting clears out the memory **51** in the switch from one user to another. Also desirable is a multi-key, multi-position lock. Any one key may turn the lock to any one predetermined position, enabling one corresponding data store **52**.

The invention now being fully described, one of ordinary skill in the art will readily recognize many changes and modifications that can be made thereto without departing from the spirit of the appended claims. For example, in addition to switching software, data stores or other

peripherals as described above, a computer may also switch properly functioning hardware for malfunctioning hardware. Indeed, in a computer with multiple mother boards, a switch may switch the functioning components of a computer from one board to another.

Also, while the description above usually uses data stores as the devices to switch, one of skill in the art will readily now realize that other computer components may be switched, including logic boards, ROM and controllers.

Under certain circumstances, danger or damage may follow from switching when power is supplied. Accordingly, a switch may be deactivated when such danger or damage may result. Logic such as the controller **1A** may prevent dangerous or damaging switching by tracking power states, device identities, etc. and permitting switching, for example, when no electrical current is flowing to the devices to be switched.

Preferably, the switch is located in an easy-to-reach location. This contrasts with the typical location of USB, keyboard and other ports, for example.

The following invention provides an apparatus and method of supporting the backup and recovery of a computing device. The computing device will typically include both a user computing environment and a supporting environment which enhances the stability and functionality of the user computing environment.

Processes

In one embodiment, a plurality of computing processes may be utilized to enable the On-the-Fly invention. Here, individual computing processes may monitor, track, predict the stability, backup, restore, or recover attributes within the user computing environment. The attributes may be software specific, data specific, operating system specific, or any combination. Utilization of the plurality of computing processes can facilitate the normal operation of the user computing environment. In one embodiment the user computing environment may be stabilized without user intervention such as requiring the user to shut-down, restart, logging off, logging on, or terminating applications. In one embodiment the supporting environment may have a capability interacting with the user computing environment. In one embodiment the supporting environment may be capable of initiating or causing the user computing environment to shut-down, restart, logging off, logging on, or terminating applications.

Different computing systems

In one embodiment the user computing environment and the supporting environment function in different computing systems. The two computing systems may reside in a common box. The user computing system may consist of data storage devices, RAM, processor, video card, and other attributes known in the art to facilitate a computing system. The supporting computing system may consist of a master template data storage device, RAM, processor, and other attributes known in the art to facilitate a computing system. In one embodiment, the data storage devices may be linked as needed to perform repairs. Such as, the need to copy data from the support environment to the user environment.

Snap-Shot of data

In one embodiment, the present invention takes a snap-shot of the user computing environment. This snap-shot may subsequently be utilized to restore, analyze, or enhance the stability of the user environment. The snap-shot may include a stable image of the operating system, software

applications, or user data. The snap-shot may contain an idealized or stable version of a disk drive utilized by the user environment, or a subset of the disk drive such as an individual partition. The snap-shot may also include an idealized version or image of the user system RAM, user system disk drive, user system partition image, memory of the video card, or any other memory stored or utilized in the user computing environment. These snapshots may be stored in the associated support environment data storage device.

Monitoring
The supporting environment may monitor the user environment. The monitoring may include monitoring of processes running or enabled within the user environment. The monitoring may include monitoring both the utilization of the data storage device, data contained on the data storage device, and other aspect necessary for the normal operation of the user environment. This monitoring may facilitate identifying undesired changes, potential problems and also potential solutions. The supporting system may detect a freeze or other undesirable change within the user environment.

Recovery
When an undesirable change is detected in the user environment, the supporting environment may attempt to recover or restore or repair the user environment. The supporting system may be capable of re-enabling the user environment in a number of ways, such as resetting the keyboard in the event the keyboard locks the communication of keystrokes to the user environment. Further recovery of the user environment may be supported by reset connections such as describe by "Freezebuster", reset and clear devices as needed, replace defective software components as needed, and/or switch hardware components and/or devices as needed. The supporting environment and or supporting system may copy all or part of the data from one or more of the idealized snapshots mentioned above. These snapshots may be copied into their respective devices and/or locations.

Application Configuration
Another embodiment supports an ability to run two or more different programs at the same time on one computing system where the data and applications may be isolated from one another but may share output and/or input devices. In one embodiment, the applications may be isolated by executing the applications in a separate address space. The applications and data may be further isolated by utilizing two separated data storage devices. In order to safely send a command from one isolated data storage device to the other isolated data storage device the following may be utilized. In one embodiment, when an icon on the desktop icon is clicked the following may occur. The icon may execute a command that would launch a specific application on the other isolated data storage device. This may be accomplished by a shared ASIC that sends the command to the other isolated data storage device.

Another embodiment involves isolation of data with merged display. In this embodiment

two user environments can be separated for the purposed of isolating data. For the Anti-Hacker System: A hard drive that does not contain "sensitive" data could be isolated and attached to a network. A second hard drive, may or may not be attached to the other hard drive (in any way), could be utilized for "sensitive" user data, but have no exposure to the network because it is "isolated" by a means of switching. The video signals associated with the data coming from these two hard drives could then be "merged" onto the same screen. In other words, all of the computing would be happening within isolated "secure zones" within a single computer but would not appear so to the user. Another example: the anti-virus system could use this method to isolate potentially infectious data.

Application Output
Applications may have its output displayed on the same screen alongside and/or superimposed upon the same screen with other applications and data that were being "computed" separately. Both computing processes may be separated but may then be "merged" together on the screen, and/or overlaid one another on the same screen. In one embodiment, this may be achieved by using multiple video cards. This concept can be applied for example to the Repair System, Multi User, Anti-Hacker, anti-theft and Anti-Virus.

In another embodiment both the user computing environment and the supporting environment will reside on a single computer system. A snap-shot of the operational user environment will be taken. The snap-shot will be associated with the supporting environment. Processes associated with the supporting environment will monitor the activities and status of the user computing environment. The monitoring function will become aware of any degraded performance of the user computing environment, such as a system freeze up. The monitoring function notifies the supporting environment of any degraded performance. The supporting environment will perform any recovery action as necessary to recover or restore the user environment. Recovery may include utilizing the snap-shot to recover or restore the user environment. An entire user disk may be restored. A specific application or software package may be restored, or particular files.

EXTERNAL REPAIR OF A COMPUTER

The invention may back up or recover a computing device. The computing device may include a user computing environment and a supporting environment which stabilizes and functionality of the user computing environment. The invention may include one or more external devices or removable media.

Master Template

A master template may be a copy of data that represents an ideal state of a computer system or component of a computer system. The master template may be created by copying data from an operational computer system or component of a computer system. The computer system may be in an ideal state before creating a master template. An ideal state of a computer system may be represented by data that is accessible to the computer system. Data, within this context, may include an operating system (e.g., Linux, Unix, Windows 98), applications (e.g., WordPerfect, Microsoft Office), user data (e.g., operating system preferences, background images, created documents), and component data (e.g., BIOS, PRAM, EPROM). Data may also include any information accessible to the computer system, including local and remote data storage devices.

As an example, the master template for one computer system may include all of the information installed on that computer system, such as Windows 98 operating system, WordPerfect application, documents created by the user. The information may be installed across multiple hard drives accessible to the computer system. Additionally, the master template may include a copy or an ideal-state version of the BIOS settings.

A master template may represent a snapshot of a newly purchased computer system. The system is typically in an ideal state with an operating system and various applications pre-installed, thereby allowing a user to begin utilizing the computer system. For a particular user, the master template may represent an ideal state of a computer system, including, for example, an operating system, applications, and user customizations. A user customization may include the users prior selection of a picture or ".jpg" image for a desktop background, such as a picture of the users pet.

Optionally, the master template may be created from a first computer system and subsequently may be used as a master template for a different computer system. An ideal

state of the first computer is thereby transferred to a second computer system or any number of computer systems.

Backups

A backup is a copy of data that represents an information on a computer system or component of a computer system. The backup may be created by copying data from an operational computer system or component of a computer system. A backup of a computer system may include data that is accessible to the computer system. Data, within this context, may include an operating system (e.g., Linux, Unix, Windows 98), applications (e.g., WordPerfect, Microsoft Office), user data (e.g., operating system preferences, background images, created documents), and component data (e.g., BIOS, PRAM, EPROM). Data may also include any information accessible to the computer system, including local and remote data storage devices.

As an example, a backup for one computer system may include all of the information installed on that computer system, such as Windows 98 operating system, WordPerfect application, documents created by the user. The information may be installed across multiple hard drives accessible to the computer system. Additionally, the backup may include a copy or an ideal-state version of the BIOS settings.

An archive is a backup which typically may not be erased.

Data Storage Device

A data storage device includes memory devices, which are accessible to a computer system. A computer system is capable of accessing or storing data in a variety of memory devices. Memory device may include hard drives, RAM, ROM, EPROM, or BIOS. Memory devices store data (e.g., data or programs). User data is typically stored on disk drives, but may potentially be stored on any memory device. Typically, a computer system utilizes a variety of memory devices. For example, an operating system, applications and user data may be stored on a hard drive, a BIOS program may be stored in ROM, and BIOS data may be stored in a protected memory.

DSD

A "DSD" refers to a "data storage device."

Methods of External Attachment

Data Storage Device (DSD) may be an external device. A variety of protocols currently exist for utilizing external devices. Some of the more prevalent protocols include TCP/IP, USB, USB 2, Firewire, IEEE 1394, PS/2, parallel, serial, PCMCIA, SCSI. Other protocols and method of connecting external devices to a computer system will be apparent to one skilled in the art. As an example, a SCSI hard disk and SCSI CDROM are memory devices that may be attached to a computer system. The computer system may then read or write to the external device.

Repair Process:

An automated process may repair a data storage device of a computer system. The repair process may include multiple programs. The automated process may be triggered by a particular event or a set of events. The repair process may be specific to a particular data storage device such as the primary boot partition of a hard drive. The repair process may encompass a variety of functions which may be modified, added, of skipped based on the type of repair or user preferences. The user may modify user preferences.

In one embodiment, the repair process represents a sequence of functions. Typically a Master Template is either provided to the user or created by the user. Backups are created intermittently . The computer system becomes unstable and repair becomes necessary. The user may activate the repair process or the repair process may recognize the instability or problems with the system and activate the repair process.

Prior to repair, a Master Template typically exists for the computer system. The Master Template may have been created in a number of different ways. Several ways of creating one or more Master Templates for this computer system include: shipped with a new computer, created with the installation of software (e.g., software to support this process), created by a user-activated program, periodically created of a Master Template by a program.

Backups typically exist for a computer system. A backup may include user data and programs which have been stored on a data storage device accessible to the computer system. For example, documents may have been created or modified by a user. These documents may be stored as a backup. The user may have installed additional programs that may be stored in a backup.

During a backup process data is copied from a data storage device of the computer system to the backup data storage device(s). Any data that is accessible to the computer system may be backed up. The backup may be compressed. Compression may reduce the amount of storage space required to hold the backup. Incremental backups may also be used. Incremental backups may reduce the time required to perform a backup and reduce the storage space required to store them. Backups may be stored as archives.

Repair Process is activated and (Optionally may be confirmed):

The repair process may include a number of functions. The repair process may be initiated by a user, administrator, repair software, or repair hardware. The user may specifically initiate the process (e.g., double clicking on an icon of a graphical operating system). An administrator may initiate the process by communicating with the computer system over an internet connection such as TCP/IP. Repair software may initiate the process by utilizing a boot diskette or a separate boot partition on the hard drive. Repair hardware may initiate the process by sensing a frozen state of the operating system or hard disk, and subsequently initiating the repair process. Alternatively, the user may press a hardware switch which initiates a process to shutdown the machine, switch boot disks, and the subsequent startup may initiate the continuation of the repair process.

The repair process may be configured to allow the user to confirm the repair process in a number of scenarios. For example, before a DSD is reformatted the user may be requested to confirm the operation. The user may be allowed to halt the repair process.

The repair process may utilize a Master Template, Backup, Archive, various commands associated with an operating system, switching, and other programs, for repairing a computer system. For example, the repair process may format and partition a hard disk using an MS-DOS command, then copy a Master Template to the primary boot partition of the hard drive, then copy the latest Backup or Archive, then mark the primary boot partition as the active partition.

Any number of backups or archives may be used to restore the user DSD(s).

Command associated with an operating system may be used to reset or update DSD of the computer system. A DSD (e.g., PRAM, BIOS, or CMOS) may be updated through the use of commands associated with an operating system. Typically, MS-DOS commands may be used to download, save, reset, reset to the default, or update a BIOS version. For

example, one step in the repair process may include booting into an MS-DOS partition, executing MS-DOS commands to update the BIOS of the computer system, then change the boot device and reboot the computer system to continue the repair process if necessary. Alternatively, the DSD (e.g., BIOS) may be set to a previously saved state. The previously saved state may be included as part of the Master Template, Backup, or an Archive.

The repair process may also be capable of managing DSDs. Managing DSDs may include testing, reformatting, analyzing, resetting, or determining bad blocks. Alternatively, the repair process may interact with other programs to provide management functionality of all or some DSDs. For example, the repair process may rely on operating system commands to format a DSD (e.g., a hard drive), but interact with a program to interact with another DSD (e.g., BIOS, PRAM).

The repair process may evaluate the present state of the computer system. As part of the analysis the repair process may determine or recommend a type of repair. For example, if the DSD (e.g., hard disk) is not responding then reformatting may be recommended. If only several files appear to be corrupted then the repair process may determine only those files need to be copied from a Master Template or a backup. Some or all of the data from a master template may be copied on to the DSD(s). Alternatively, the repair process may copy the entire master template to the DSD(s).

The repair process may perform a similar evaluation regarding how much of a backup needs to be copied. Some or all of the data from a backup may be copied on to the DSD(s). Alternatively, the repair process may copy the entire master template to the DSD(s).

Rebooting the computer system may be integrated into the repair process. Switching between boot devices may be integrated into the repair process. The repair process may switch the boot disk from hard disk 1 to hard disk 2. Power may be cycled such that hard disk 2 boots up as the active partition. A default program may be executed as part of the boot sequence to perform part of the repair process. Subsequently, the repair process may alter the hard disk 1, switch hard disk 1 to the active partition, and then reboot or cycle the power to initiate the booting of hard disk 1.

Some Examples of External Device Embodiments

The repair process may be initiated or managed by an externally located device that

may be communicative coupled to the computing device through, e.g., USB, Firewire, parallel, serial, PS/2, PCMCIA, or infrared. The external device may be the boot device.

An external boot device may be connected to the computer system with the boot device activating the repair process. The repair program may reside on the boot device or a second data storage device. The second data storage device may also be communicatively coupled to the computer system. The second data storage device may contain master templates, backups, or archives. The second data storage device may also contain the repair program or other programs which facilitate the repair process.

For example, an internal SCSI device "id 0" may be the default boot device. The repair process may switch the power to the SCSI device "id 0" OFF. The repair process may switch the power to an external SCSI device "id 0" ON. The repair process reboot the computer system by actuating a reset command (e.g., a mechanical device, a logic circuit). When the computer system reboots, the external SCSI device may be the boot device. The repair process may then continue as directed by part of the repair process on the external SCSI hard drive.

The repair process may include switching the device id's of a primary and secondary SCSI disk. In this second example, the internal SCSI drive may be "id 0" and the external SCSI drive may be "id 5". The repair process may change the internal SCSI device to "id 5" and the external SCSI device to "id 0". Switching of the SCSI device id's may be performed by the repair process (e.g., a mechanical device or a logic circuit, activated by the repair process).

In another embodiment, the BIOS may be modified to enable booting from an external device. The boot device may also be switched by updating the BIOS. Typically the BIOS defines the boot sequence. If the first boot device is not found, then an alternate boot device may be defined in the BIOS (e.g., the boot-device sequence is CDROM, A:, C:). The BIOS may be downloaded, modified, and restored. The BIOS may be updated (e.g., in place, via download-modification-upload) to change the boot identifier of a USB device, an IDE device, or other devices. The repair process may download a copy of the BIOS in a variety of ways. One example, includes booting into an MS-DOS mode, executing a program to save the current BIOS to a file. The BIOS file may be saved into a master template, backup or archive. Alternatively, the BIOS file may be modified by the repair process to change the boot sequence. If the BIOS file is updated then it must be

loaded into the computer system to take effect. Effectively the boot sequence may be changed to another DSD, such as a second hard drive. The external SCSI disk with a specific "id" may become the "boot device". Another option involves storing multiple copies of the BIOS file, each having a different boot sequence, uploading the appropriate BIOS file may allow booting from a particular boot device (e.g., IDE hard drive partition 1, SCSI device "id 0", USB disk, Jaz drive, etc.). An external device may be the boot device and start or continue the repair process.

In another embodiment, a secondary boot device may be attached as an external Data Storage Device to a computer system (e.g., connect to a parallel port). This secondary boot device may activate or manage the repair process. The secondary boot device may contain programs to conduct processes such as reformatting another data storage device (e.g., internal or external hard drive), copying data from a Master Template, copying data from a backup or archive.

A program on the secondary boot device, or accessible to the secondary boot device, may be activated to create a master template, backup, or archive of and data accessible by the computer system (e.g., the user's main drive).

A program on the secondary boot device, or accessible to the secondary boot device, may be activated to repair a data storage device on the computer system (e.g., the user's main drive that needs to be repaired). In this scenario, the Master Template, Backup, or archive Data Storage Device(s) may be attached externally via USB, firewire, etc. The program may actively search for Master Templates, Backups, or archive DSD(s) and present the user with a list of options for restoring the computer system. Alternatively, the repair process may determine and select the best restore options and continue the repair process.

In another embodiment the repair process may be initiated by insertion of a floppy, cd, dvd, or use any other form of removable storage/memory or startup device, and rebooting the computer system. The removable storage/memory or startup device may boot if the BIOS boot sequence contains a sequence in which the boot order enables a removable media to act as the boot device. Booting from the removable media may trigger or activate an automated repair process (e.g., a program located on the removable media or an external device). Booting from the removable media may activate a mechanical device or program logic to initiate the repair process (e.g., switch hard disk device id's and initiate a reboot

sequence to boot from another device to continue the repair process).

In another embodiment, a repair program or part of the repair process may be placed in a StorExecute, microcontroller, ASIC, etc. The repair program may activate a repair process. The repair program may include managing the repair process. Functions which may be performed include reformatting data storage device(s), switching between boot devices, switching electrical components within the computer system or external components, copying data to/from data storage device(s), (e.g., copying master templates, backups, etc, or any portion to another data storage device), and other repair functions. The repair process, may also be located, integrated, or embedded in an external device. A switch trigger that activates the repair process may also be located, integrated, or embedded in an external device.

In one embodiment, the startup device may be selected by a StoreExecute. Alternatively, a device identity may be assigned by a StoreExecute. The necessity to perform switching through the use of jumpers is thereby reduced. For example if a repair process is triggered, a StoreExecute may assign device identities to data storage devices or may decide which data storage device shall be used for the repair process, and which data storage device shall be used for boot data storage device if rebooting is utilized in the repair process.

In one embodiment during "on-the-fly" repairs, an external data storage device may be utilized for such things as the Master Template or backups, or for software used for the repair process.

In this embodiment, an external data storage device ("DSD") is attached to a typical personal computer that contains an internal data storage device. The internal DSD may be referred to as the "main user" data storage device. An external DSD may be attached via any available external connection.

Example of external data storage device ("DSD") for repairing a computer:

In this example, a user attaches an external data storage device ("DSD") to a computer with any available external connection (e.g., Firewire, USB, SCSI, etc.). An external connection may include USB, USB 2, Firewire, IEEE 1394, PS/2, parallel, serial, PCMCIA, SCSI, and other protocols and method of communicating with an external device.

The user installs software on "main user" DSD that initiates a program to create a master template, and schedules Backups to execute every Friday morning. The master template is created by the program and stored on the external data storage device. Every Friday morning the repair process runs and stores a backup of additional information to the external data storage device.

A micro-controller and EPROM may be attached to the computer to perform part of the repair process. Attachment may be via any available external connection. The micro-controller and EPROM may be integrated into the external data storage device.

A switch trigger may be attached to the computer. Attachment may be via any available external connection. The switch trigger may be integrated into the external data storage device.

As another example, the main user data storage device is accidentally erased or damaged and that the computer system will not boot. The user decides to repair computer and initiates the repair process by activating a switch trigger, which initiates the following process:

The micro-controller may interrogate the BIOS of the computer system to determine its current boot up sequence. EPROM may store instructions for how to accomplish this.

The micro-controller may determine that it is necessary to alter the boot sequence so that the externally attached data storage device will become the boot device. The micro-controller and associated EPROM may flash the BIOS in order to accomplish this. The micro-controller may then send a command to computer to reboot the computer. When the computer reboots, it will reboot from the external data storage device.

Following the boot up, programs which are located on the external data storage device may execute the repair process as defined herein.

BACKUP OF A COMPUTER

The invention may backup, maintain backups, or recover data associated with a computing system. The computing system may include any number of components including hardware and software, and any memory accessible to the computing system. The computing system may focus on a user computing system and potentially the supporting environment which stabilizes the functionality of the user computing system (e.g.,operating system, BIOS, etc.). Typically data associated with the computing system is identified by a variety of characteristics, the data is stored as a backup, and subsequently data within the backup may be restored or used to evaluate an existing computing system.

Backups

Data has a number of characteristics, typically including availability for use in a computing system. Data may include one of more of any of the following: operating systems, application, user data, data residing in the computing system (e.g.,hard disk, hard disk partition, RAM, ROM, BIOS, CMOS, EPROM, electronic serial numbers, etc.), applications residing in the computing system (e.g.,sample listed above), and backups created or accessible. The term data may be used to describe a specific aspect of information for association with a backup process. A backup process may include identifying data and the characteristics of data, for backup, management, or restoration. Data may also refer to a backup or set of backups. By default the data to backup may represent all data on a given disk drive, a given disk partition, or a memory.

Characteristics of the data may include an indication of what data is part of the backup, how to access the data, where to backup the data, frequency of the backup, and type of backup. These characteristics may be used to define or identify specific data associated with a backup process. Specific implementations may vary according to what characteristics are associated with the backup process.

What data to include is limited by the accessibility of the data to the computing system. Specific data for inclusion in a backup may be predetermined or determined as part of the backup process. Predetermined identification of data to include in a given backup may be provided by a hardware or software manufacturer, or a user (e.g.,system administrator). Predetermined set of data may provide an initial indication of what data to backup. An operating system may, for example, include a list of files and or directories

associated with operating system functionality. Here the operating system may provide a predetermined list of files or associated data representing the operating system or identifying specific data to backup (e.g.,list of uses, user preferences, passwords, windows registry file).

A hardware system may, for example, include a memory address range (e.g.,RAM, ROM, EPROM, BIOS, etc.) that represents data that may be useful to backup for that system. The hardware system may also identify other data within the computing system that may be useful in the backup process (e.g., applications to extract or update a BIOS). Typically, the data identified is useful in the backup process, such as understanding the operation of the computing system or restoring data in the event of a failure or corrupted data. Data identified for backup may also have a variety of uses including cleaning up the computing system which may have limited disk space (e.g.,verify the necessity of data in a current computing system) and restoring identified data.

Alternatively, what data to include in a given backup may be determined subsequent to the delivery of a computing system to a user. Data may be determined with installation of hardware or software, or during the normal course of utilizing the computing system. A determination may be made with the installation of hardware or software. The installation process may be actively engaged in identifying what data would be useful to the backup process. The installation process may interact with the backup process or tools to identify program files and data specific to a given installation. The location of user file may also be helpful to the backup process. The contents of a user directory may be marked by the backup process for inclusion in a periodic backup. Accessing data by an application may also be integrated into the backup process. One example includes added functionality, such that saving data (e.g.,a files) by the application includes an indication to the backup process to backup that specific data. The installed application may add the saved user file to a list of files that should be include in a subsequent backup. If multiple users access the same computing system, the file to be included in a backup may include an ownership indication.

Data to include may be identified according to directories or specific files. For example, data to include may be identified by file type, file location, directory tree, of memory device. A selective backup may backup only data associated with a specific system component such as a disk drive or data storage device.

How to access the data may be an important characteristic of the backup. An

important consideration may be required for accessing, storing, formatting, modifying, restoring, and updating data of the various components associated with a computing system. Not all data is readily accessible according to a well known process of accessing a hard drive. As described above, data may include any data accessible to the computing system. Typically, a piece of data is uniquely accessible according to a predefined process. The process for accessing information from a disk drive is readily appreciated by novice users.

For example, accessing BIOS data for backup may involve booting into a particular operating system (e.g.,DOS 5.x), running a hardware-specific program which may verify the hardware compatibility, executing a second hardware-specific program which may copy the data (e.g.,BIOS data) to a floppy disk. Updating the BIOS in the example may involve running another program to flash the BIOS. Both the old and new versions of the BIOS, and associated applications can be stored as data in a backup. Consequently, a restoration of the old BIOS can be incorporated into the backup process. Similarly, other data accessible to the computing system may be incorporated in to the backup process by analyzing the existing processes for managing data for specific components within the computing system.

Where a backup is stored may be predetermined or determined as part of the backup process. A manufacturer of the hardware or software may provide an initial predetermined backup storage area or an indication of another device where the backup is to be stored. An operating system may access a second data storage device such as a disk drive, a second partition, or a pre-allocated file (e.g.,similar to a swap file). Backup data may be stored to this initial location. A Hardware system may, for example, include a second memory or an address range of a memory (e.g.,RAM, ROM, EPROM, BIOS, etc.) that represents the default backup location. Optionally, the backup location may be another storage device within the computing system or accessible to the computing system (e.g.,across an Ethernet, firewire, USB, etc.).

Frequency of the backup can be based on any of a number of factors associated with the data and computing system including: volatility of data, volatility of the computing system, importance, upgrade schedule, user projects, personal comfort level, past experience with similar environments, degree of user participation, etc. Backups can be scheduled at particular times and intervals based on these factors. Backups may be initiated

by the hardware, software, or a user. Similarly, other activities on the backup process, such as maintenance and restoration, may be performed based on a given frequency.

Type of backup

A variety of backup types may be supported. The types may include at least one of the following: full backup, selective backup, partial backup, master template, data modified since a prior backup, or based in part on a comparison with a prior backup (e.g.,a prior backup, or a listing of the contents of a prior backup). The type of backup may be defined for all data included in the backup, or part of the data associated with the backup process. For example, a backup may include an operating system wherein only files associated with the operating system and files modified since a prior backup are included in a specific backup. The specific backup may further include a user data directory identified for backup.

Data represented in a backup

Data represented in a backup may be identified by the various characteristics described above. Typically, data represented in a backup supports a backup process, such as a possible restoration of the data for use in a computing system. The backup or the various data contained in the backup may be a compressed or encrypted. Specific data in the backup may be an exact duplicate or enough information that the data may be recreated, corrected, or verified. For example, file differences may be included in a backup, thereby allowing a set of backups to be utilized to recreate or correct a file or data. How to access the data may also be represented in a backup for certain types of data (e.g.,BIOS) and not represented in a backup for other types of data (e.g.,"c:\my docs*.docs").

Data to be included in a given backup may identify by hardware, software, user, or other characteristic of the computing system. A computer manufacturer may create an initial backup of a standard installation, which may include various forms of data associated with a computing system. The manufacturer sells the computing system to a user and may provide a master template as a backup that represents the manufacturers initial computing system configuration. This saves the manufacturer time and money, and gives the user peace of mind. Subsequently the user may install additional software and thereafter create a

partial backup of the changes to the computing system. A comparison may be performed between the master template and data associated with the current computing system. Difference between the two can be identified as the data for backup. Here, data that has been changed, added, or deleted, in comparison to data associated with a master template may be identified for backup. Consequently, the master template and a subsequent backup may be used, according to this example, to restore the computing system to the level of functionality associated with the subsequent backups. A variety of scenarios will be apparent to one skilled in the art.

Repair Process

Restoring

Data represented in a backup is typically restored to a computing system. Restoration may include the selection of at least one of the following: specific backup, group of backups, specific data contained within a backup, and a master template. The restoration may initially determine the difference between the current computing system and a prior backup. Characteristic associated with the identified data may be used in the backup process (e.g.,restoration process associated with BIOS which may have been included in a backup.).

The selection of a master template, for example, may return the computing system to an idealized state as defined by the master template. A master template and other data may be identified to restore the computing system to a state associated with the last backup in combination with the identified master template (e.g.,master template represent the state as purchased, and the identified backup represents the state after a user installed several applications). Alternatively, a master template may represent an upgrade to the computing system. This upgrade may be combined with other user backup to enhance the functionality of the computing system and maintain existing user data.

Selecting Data

Data associated with the backup may be identified similarly to the selection of data for inclusion in the backup, as described above. This information may also be utilized to determine what data or aspects of the data to restore (e.g., specific users files).

Data matching a certain file type, file location, data storage device, device,

component, description, date, wild card matching, etc. may be identified for restoration. The selection may be performed by the hardware, software, user, or any component in the computing system. In the event of an operating system failure it may be more appropriate to allow hardware or software select data to restore.

Restoration location for data may be specified by a user, hardware, software, default, original location of the data, temporary location, an alternate location (e.g.,for further analysis), or by any component of the computing system. For example, a user may elect to restore data with wild cards such as "*.doc" and "*.txt" from all backups. The "*.doc" files will be placed in a user-specified or default file location (e.g.,"c:\documents folder\doc\"), and "*.txt" files will be placed in a user specified file location (e.g.,"c:\documents folder\txt\"). Alternatively, the data (e.g.,files in this example) may be restored to their original location which may be identified in the backup.

Preferences

Preferences may be associated with the backup process, and may include preferences of hardware, software, users or other components of a computing system. Preference may be defined as a set of default values associated with the computing system, hardware, software, or particular users. Configuration information and characteristics may be defined as preferences for each component of the computing system. A preference associated with a BIOS may include a process or program for accessing the BIOS in a specific manner, such as booting to DOS 5.x, executing a specific program to extract the BIOS. Preference may be changed by hardware, software, or users.

The preferences can be used to define data characteristics (including backups), restore characteristics, and manage data. Preferences may limit the interaction required with users during the backup process (e.g.,selecting data or restoring data). A new user may establish preferences to limit interaction with a backup process. A seasoned veteran may establish preferences to provide a more robust control of the backup process or aspects of the backup process.

For example, the specific characteristics of how the backup process interacts with updating a BIOS may be of a greater interest to an experienced user rather than a novice. In another example, user preferences may dictate the interaction between the user and the restore. By default, the restoration process my provide the user with a push button restore,

such that the computing system will control the entire restoration process. Alternative, the user may modify the preference such that a user response is required before specific aspects of the backup process are performed (e.g.,format hard drive, or flash the BIOS).

Software may also have preferences, which may identify data associated with the software, when installed, serial number, and possibly an indication of the best way to backup, manage, and restore the software. Preferably, preference associated with hardware and software would minimize interaction required a by user in the process.

Initiating Restoration

The hardware, software, or user may initiate and may manage the repair process. Data matching a restoration criteria may be restored. Criteria for restoration may be base on the data stored in the backup (e.g., frequency, master template, compression, encryption, etc.). Further criteria for restoration may be based in part on the type of backup or current status of the computing system (e.g.,functional, hard disk failure, BIOS failure, OS non-responsive, etc.) The current status may be determined in part through the utilization of hardware and software to monitor the health of the computing system. For example, hardware or software can monitor the computing system for any indication of a keyboard "freeze", and activate part of the backup process to return the computing system to a normal operating state. Utilization of hardware and software can be used to maintain the health of the computing system. Maintaining the health of a computing system may include determining backup process characteristics which may be based on user preferences. The frequence of backup may be a way to help ensure the computing system's health.

For example, an alternate boot sequence may be initially established in the BIOS such that the computing system initially attempts to boot from a primary disk drive and subsequently to a second drive. The second drive may contain software designed to boot the machine and evaluate the present condition of the computing system. Once the necessity of any repairs have been determined, the software may proceed to correct the malfunctions and return the computing system to a normal operating state. The software may then reboot the computing system to the normal operating state, thereby minimizing user involvement in the repair process.

Removing Data

During a restoration, process data may be removed including: deleted, moved, renamed, or altered. The method of removal may be specified as part of the data characteristics. The restoration process may require the computing system to reflect the data contained in a backup, and therefore necessitate the removal of some data. For example, in restoring data representative of an operating system, a preference may provide that existing inconsistent files may represent the culprits behind a malfunction predicating the restore process. Removing this additional data (files in this example) may be warranted. Removing extraneous data may be performed in a number of ways based in part on the type of restoration, preferences, characteristics of the backup or data, and the goals of the backup process (e.g., minimal user involvement). For example, if the goal is to restore the master template, then as part of a comparative restoration all data determined to be different from the master template may be removed to a specified data storage device or memory such as a default folder.

Restore Specific Data

The hardware, software, or user of a computer system may request the restoration of data. To facilitate the restoration of specific data a user may perform a restore based in part on: file type, creation date, user identification, modification data, backup date, or any characteristics of the data. For example, a completed restore may include a default folder that contains all data from the last backup which differs from data currently available for access to the computing system or some subset of all of the data (e.g.,specified according to preferences). Alternatively, the folder may contain all data which differs when comparing two backups, such as the last backup and a master template. Data conforming to the users request may be sorted into different directories to provide the user with an indication of the information contained therein, such as "This is probably your stuff 2/25/03", "Is any of this your stuff? 2/25/03", and "Probably not your stuff2/25/03".

Managing Restored Data

Preferences may also control what happens to restored data. Data restored may be available to the user or the computing system for a limited duration, to reduce the amount of memory utilized by the computing system. For example a user definable preference may indicate that a dialog warning that the folders named "Is any of this your stuff?2/25/03" and

"Probably not your stuff2/25/03" will be automatically deleted in 10 days and if the user desires data from those folders the data should be moved prior to the expiration date. Optionally, a preference may provide that after 10 days the contents specific folders may be moved to a temporary "trash" folder with a new expiration date of 30 days.

Placement of restored data

Placement of data may be defined in part by the data characteristics stored with the backup or data, the characteristics associated with the backup process, and the preferences. Data, such as user data, may be returned to an original location, and other data may be placed in a different location. For example, user data located on the desktop may be returned to where it was, whereas user data located in the system folder may be returned to its original location depending in part on preferences. Alternatively, user data may be deposited in a default or indicated location such as a "documents" folder, a "Your Stuff is In Here" folder, a "proposed trash" folder, a "trash" folder, or other custom locations.

Master Templates

A master template is a backup of data, representing a computing system according to an ideal state. The ideal state typically includes an operating system, a collection of applications or software. The data included in the master template may have been specifically chosen for a particular user and for a particular hardware configuration.

A master template may be created or updated according to a variety of approaches. One approaches involving a data storage device may include: 1. Creating several backups of data on a data storage device over time; 2. An activity associated with the backup process, such as a repair process is triggered; 3. A backup of user data files is performed (e.g.,to save the users current work) ; 4. Existing data storage device (e.g.,memory) may be reformatted or tested, and may be performed according to preferences for that data storage device; 5. The master template is copied to the user data storage device; 6. Backup of user data files is restored to the user data storage device. 7. The computing system is thereby restored to a normal operating state with minimal user intervention.

The master template may also be updated, changed, or modified in a variety of ways including: by the user, by access to an update (e.g.,an incremental release by a computer manufacture), or by access to a replacement master template, etc. The preferences

associated with a master template may provide a method for performing these modification.

The master template may be tested to ensure the master template and the repair process functions as expected in the backup process, such as restoring the computing system This testing helps ensure the functionality of the master template, the restore process, and may also be used as a virus check and repair. An on-line service may be provided to detect virus, verify the integrity, or to update a master template.

Restoring

A backup may be tested to verify its integrity (e.g.,with a checksum and verifying readability). If the backup is tested and fails, the user may change the preferences. The user may restart the repair process, select different preferences (e.g.,applications or software), upgrade the backup (e.g.,master template), and retest the backup. If the backup passes the verification tests, the user may accept the backup and continue with the restore. When a backup (e.g.,master template) is accepted it can be copied from its storage location to a second backup (e.g.,the new master template). The old master template(s) can be saved so that it is possible to revert back to prior master templates. After the user template is "accepted", the backup user data is returned to the user data storage device.

In one embodiment, a master template can be created by the user selecting to "boot into" a master template. The user may then make changes, install new software, make modifications, etc., and then exit. This approach allows the master template to be updated independently of user's documents and other data which may not be a beneficial to a master template.

In a different embodiment, the master template may be modified/updated by the user first conducting a repair of computing system. The repair process may automate 1. The backup of user files according to preferences, potentially including particular file types (e.g.,documents); 2. the reformat of the user's primary disk drive or the restoring of the master template to the user's primary disk drive. The user may then install new software to an essential copy of the master template as present on the user's primary disk drive. A backup may subsequently be activated to generate a new master template version. A backup of the user's data (e.g.,user specific documents) may then be restored to the computing

system. Preferably, restoring the user specific documents is performed automatically.

The master template may be created by a process of selective copying. For example, depending on the particular OS in use, a program may interrogate the registry, determine what entries are associated with a particular program or application, and then choose to selectively copy only those files and entries associated with the particular program or application to the master template.

A COMPUTER WITH SPECIAL-PURPOSE SUBSYSTEMS

Switching mechanisms

A variety of events may trigger a repair system to perform a repair process on a primary system(s) to be repaired. An event, such as switch triggers, may include single step and multiple steps. Each step may include a logical or physical action initiated by the repair system itself, user, external system, or the primary system to be repaired. A step may include a logical or physical confirmation of the repair process. Individual steps may be automated by the repair system, switching process, or a primary system. An example of multiple steps that trigger the repair system may include 1) pressing a button, and then 2) sliding a switch for confirmation of the repair process. Other steps will be apparent to one skilled in the art and are therefore not described herein.

The repair may include any process that attempts to place a primary system into an idealized state or restored state. The repair system may include various apparatuses and methods previously described, including the switch process. As an example, the repair system may be triggered by voice recognition or voice identification associated with an individual step or multiple steps of a triggering event. In one embodiment, pressing a physical button triggers the repair process.

In another embodiment, the repair system may include a processor and logic that is independent from the primary system. Events may trigger the repair system independently of the primary system. The repair system may be triggered by a variety of events independently of the primary system to be repaired. Here, the repair system would be capable of receiving or recognizing the triggering event.

For example, the primary system may be nonoperational while the repair system remains operational with the capability of recognizing events that trigger a repair process, such as a user request to repair the primary system. The repair system may perform the repair process or may trigger

another system or application to perform the part or all of the repair process. Other applications may include such programs as: Virus Scan, Virex, Arcserve, Assimilator, Deep Freeze, Ever Dream, Filewave, Ghost, Goback, HddSheriff, PCRdist, Retrospect, RevRdist, Rewind, Hard disk toolkit, Anubus, Drivesetup, and Charis Mac.

A repair system may include a physical switch used as a step of a triggering event for a repair process supported by other applications. Alternatively, the triggering event may activate a repair process that is performed by other applications. For example, steps associated with a button, voice command, personal identification card, retna scan, or push button with a confirmation by a slide button, key switch, or diagnostic process, could be used to activate a repair process by other applications.

In another embodiment, when a primary system, such as a computer, is started an application associated with the repair system may be triggered to perform diagnostics on the computer. The application may be used to determine if the second computer attempts to start, such that, if the second computer does not attempt to start then the repair system may modify the boot sequence of second computer to boot to a different device. The application may also initiate the rebooting of the primary system. If the second computer does start, the repair system may analyze or record the boot sequence. If boot sequence fails, the repair system may automatically reboot the primary system using a different data storage device to boot and may also initiate the repair of the primary system. The repair system may also manage an "on the fly" repair process, as defined previously.

In one embodiment the push of a button (or other trigger event) triggers the repair system to perform a diagnostic process and based on diagnostic results the repair system may perform the appropriate repairs. Physically pressing the button may be the only step of the triggering event. As part of the repair process, the repair system may perform a diagnostic

process. The repair process may include interacting with a user to determine the repair process. For example the user may be prompted to respond to several questions, such as, "Your computer will soon need a repair that could take 60 minutes to perform, alternatively a temporary repair may take 5 minutes to perform. Which repair should be performed?" The user response may be taken into consideration by the repair process.

A computer with multiple special-purpose subsystems

This section provides apparatuses and methods of protecting computers and computing devices from hacking, viruses, cyber-terrorism, and from potential damage or intrusion such as spy software, keystroke recorders and damage from hacking, viruses, worms, trojan horses, and similar threats and vulnerabilities. Cyber-terrorism is an attempt to cripple or subvert a computing system. The present invention provides a solution to potential cyber-terrorism.

A computer system of the prior art typically includes: a processor, memory, display, a display controller, and input/output controller. The present invention provides a plurality of special-purpose subsystems housed within a computer system. These special-purpose subsystems typically perform limited functions and have limited interaction with other special-purpose subsystems.

Special-purpose subsystems may be designed for many purposes, including to support storing information, performing work, and handing communication. A storage special-purpose subsystem may be designed to store data and retrieve data, while allowing limited assess to the stored data. A working special-purpose subsystem may be designed to process information, such as a general purpose computer with various applications. A communication special-purpose subsystem may be designed to facilitate communication between other special-purpose subsystems.

Each special-purpose subsystem typically includes: processing

capability, memory, logic, and an interface. Processing capability may be a computer processing unit (CPU) or ASIC. The processing capability may be the computer-system CPU, or a CPU shared by multiple special-purpose subsystems. Thus, the processing capability associated with a special-purpose subsystem may also be used by the computer system or other special-purpose subsystems.

Memory may include any data storage device accessible to the special-purpose subsystem. Further, a specific memory area may be divided into logically separate areas, each of which can be associated with different special-purpose subsystem. A controller associated with the specific memory area may be configured to restrict access of a given logical memory area to a specific special-purpose subsystem. Each specific memory area may thereby be effectively isolated for use by a special-purpose subsystem.

The logic of a special-purpose subsystem supports the intended function of the system, such as storage, work, or control. The logic may include the ability to move a file, display a file, provide a directory of information available from special-purpose subsystem and other functions as necessary. Further, the logic may include or be incorporated in an operating system associated with the special-purpose subsystem. The logic may be read only or inaccessible from other special-purpose subsystems to avoid potential attacks. For example, the logic may analyze and record when files are read or written, access attempts, and associated timing. This information may be used by the logic to determine if protective measures are necessary, such as prompting the user for a confirmation of an action or denying access to the special-purpose subsystem.

The interface of a special-purpose subsystem supports the intended function. An interface of a storage system may include logic to read and write files. An interface of a working system may include a copy of a master template and applications to process and modify information,

including storing temporary files. A controller system may provide an interface for receiving requests from a working system, requesting a file from a storage system, receiving the file from the storage system, and sending the requested file to the working system.

A interface may also support interaction with common controllers of the computer system, such as for a display, keyboard, or mouse. Alternatively, the special-purpose subsystem may include a separate controller for accessing common peripheral devices. Each of the interfaces associated with a special-purpose subsystem may be enabled or disabled according to a logical or physical switch, such that interaction with the special-purpose subsystem is halted or restricted to a subset of functionality associated with the interface.

According to one embodiment, two special-purpose subsystems are provided within a computer system, the first being a working system and the second being a storage system. The computer system may include a display, a display controller, and an I/O controller. Both of the special-purpose subsystems are capable of interacting with the computer system display controller and the computer system I/O controller. A separate area of the computer-system display may be associated with each of special-purpose subsystems. If a display area is selected or otherwise active, then keyboard, mouse or other I/O-controller-mediated input would be accessible to the associated special-purpose subsystem.

Another embodiment, includes a working system and a storage system that does not allow execution of data stored (with the exception of the storage-system logic). The storage system prohibits the execution of user data, such as any information stored by a user in the memory of the storage system. The two systems are isolated from one another, and therefore events taking place in the working system cannot directly affect information stored in the storage system. Communication of data between the two systems may be through a communication controller that performs

a copying process associated with moving data, such as a file, between the storage system and the working system.

Communications between special-purpose subsystems, such as the working system and the storage system may be through a communication controller, according to one embodiment. The storage system may communicate specific information to the communication controller to transfer the specific information to the working system. The communication controller may also transfer specific information from the working system to the storage system.

A user selection of a file in the storage system can be used to prompt a communication controller to copy the file from the storage system to the working system. The file can be executed or processed in the working system. Then, the file may be saved causing the communication controller to copy the file from the working system to the storage system. In the storage system the file is not executable and thus could not corrupt other files or data associated with the storage system even though the file itself may be infected with a virus or corrupted. The working system does not typically allow user data, e.g., document files, to be stored in the working system unless they are currently being used, e.g., temporary files.

Alternatively, the communication controller may interact with the common controller to display information available from the storage system. User selection of the specific information may be performed through interaction with the communication controller. For example, the communication controller may request a list of available files from a storage system, and arrange them for a display of the list through a common display driver. A user could select a file from the list for processing in a given working system. Consequently the communication controller may cause the file accessible to the storage system to be copied to the given working system. After the working system is finished processing the file, the file could be saved through the working system's interaction with

the communication controller. As such the storage system and the working system are not required to directly interact with one another.

Additionally, the communication controller may preform an analysis on data accessible or transferred by the communication controller to determine the level of threat associated with storing or transferring the data, may refuse to handle the data based in part on the level of threat, may present the user with information which indicates a threat and a request to confirm the transfer or storage. Information presented to users may include the number of requests in a given time frame, extent of modifications, or origination location. The user response may be received by the communication controller and used to determine whether to allow the transfer or storage.

The working system may include a copy of a master template that represents an idealized state of an operating system. The working system may be an existing computer system capable of running an operating system, and additional logic for interaction with a special-purpose storage system. Typically the working system is incapable of interacting directly with the storage system. According to one embodiment, an interaction may be initiated by the storage system, or the controller system.

The working system is a special-purpose subsystem, and may be used to perform processing, editing or modifying data. The working system typically includes logic to display information to a user through the display controller to the computer display. Users can interact with the working system as though it were the primary computer system. The display controller and I/O controller may be used by the working system to interact with other devices associated with the computer system.

The storage system is a special-purpose subsystem, and typically includes data files that are stored in a data storage device. The data storage device may be volatile or non-volatile. The storage system may represent an existing computer system capable of running an operation

system, and additional logic for interacting with a working system.

According to one embodiment, the storage system initiates an interaction with the special-purpose working system. Alternatively, the storage system interacts with other special-purpose subsystems through a communication controller. The storage system may include logic to display information to a user through the display controller coupled to the computer display.

Each special-purpose subsystem may present information to a user by utilizing the same computer display. Thus, information presented on the computer display may overlay other information being displayed by another special-purpose subsystem. The user may select specific information, e.g., a document file, to work on. The user selection of the specific information may be communicated to the storage system through a common device associated with the computer system, such as a serial I/O controller connected to a mouse or keyboard. The serial I/O controller may be utilized when storage information is presented to the user. After specific information is requested, the storage system may transfer the specific information to another special-purpose subsystem such as a working system. The storage system may initiate the transfer of the specific information. In one embodiment the storage system initiates the transfer to a working systems interface. Alternatively, the storage system initiates the transfer to a common memory area for access by a working system. Another embodiment provides the storage system transfers the specific information according to a communication controller to the working system.

The working system may then access the specific information provided by the storage system. After processing, modifying or viewing the specific information, an altered version may be saved or returned to the storage system. Before saving the specific information, the working system may preform an analysis to determine the level of threat associated with

storing the information, and may refuse to save the information or may present the user with a confirmation request and information which indicates a threat. The working system may save the specific information to the storage system, the working system may transfer the specific information to another special-purpose subsystem such as a storage system. The working system may initiate the transfer of the specific information. In one embodiment the working system initiates the transfer to a storage systems interface. Alternatively, the working system initiates the transfer to a common memory area for access by a storage system. In another embodiment, the working system transfers the specific information through a communication controller to the storage system.

In one embodiment, the storage system may preform an analysis to determine the level of threat presented by storing the information, and may refuse to store the information or present the user with a confirmation request and additional information which indicates a threat.

Data may be moved between special-purpose subsystems using a separate logic control device, such as an ASIC or logic control device utilizing direct memory access. The process of moving data does not allow the data to be executed, which could possibly enable hacking, viruses, and the like. Additionally, data may be encrypted, compressed, or encoded to prevent its execution.

A control system may be an additional type of special-purpose subsystem, and could provide overall operation of the computer, computing devices, and other special-purpose subsystems. Additionally the control system may orchestrate the process of copying data, switching network communication, and repair functions as needed. The control system may be read-only, permit read only access as needed when interacting with other special-purpose subsystems such as a storage system or working system. Both the network communication and repair process may be controlled by the control system. Optionally the control system

could have limited communication with other special-purpose subsystems while maintaining an ability to initiate or conduct a copy process, activate and terminate communication to other special-purpose subsystems.

Special-purpose subsystems may be combined into a single special-purpose system that performs functions associated with the individual special-purpose subsystems, such that the single special-purpose subsystem functions performs the functions as separate threads. In one embodiment, a storage system, communication system, and working system may be combined into a computer system as individual processes executed by the computer system. The computer system may utilize any method of isolating the individual processes using techniques known in the art.

In contrast, a special-purpose subsystem or a set of special-purpose subsystems may be spread out over a number of additional special-purpose subsystems, such that some of the functionality associated with the system or set is performed by the additional special-purpose subsystems.

Repair Process

Optionally, a special-purpose subsystem may be repaired or returned to an ideal state using an automated repair process. Such repairs may be conducted "on the fly", or after each transaction or without rebooting. Master templates typically represent an ideal state of a special-purpose subsystem, and may be stored on a storage system. A transaction may include reading e-mail, wherein the opening of each individual e-mail messages represents a separate transaction. Optionally, one or more items can be ignored during a repair process. For example if an e-mail has been opened, a repair process may run ignoring the open e-mail, detect and repairs problems, and then a user may respond to the e-mail without quitting it. In another embodiment, all downloads and e-mail can be

saved immediately to the storage system prior to opening the download or e-mail in the work subsystem.

In one embodiment, the logic of a special-purpose subsystem, such as a working system, may trigger an event associated with a repair process. The repair process may perform a comparison between a master template of the working system and state of the current working system. Any differences between them could trigger a subsequent repair process in which some or all data that is different is deleted from the working system. Further, data may be copied from the master template by the repair process as necessary. In one embodiment, the repair process may make the working system identical to the master template.

In one embodiment, a repair process can be conducted after one or more e-commerce transactions, or after surfing one or more web pages, and the like. Thus all known and unknown viruses and trojan horses can be made impotent prior to the next transaction. While this process does not eliminate viruses, worms and trojan horses from the computer (they may be stored in the storage system), it keeps them in an in operative state. The repair process could repair volatile and non-volatile memory, or clear volatile memory, or set volatile memory to an ideal state.

In one embodiment, if the user selects more than one e-mail to open, two or more e-mails could be copied to the working system and could be open simultaneously. Optionally each e-mail could be copied to its own separate isolated working system, opened, viewed, and worked on separately. If the user needs to copy data from one isolated e-mail to another isolated e-mail, a copying process can be used that does not allow code to execute.

In one embodiment, web commerce software, or e-mail software, or any software can be modified so that individual records, or only copies of the records that are specifically needed for a transaction are copied to the storage system, utilized and then copied back to the database in the

storage system, and after each such transaction a repair can be conducted. Optionally, in a transaction in which data interacts with more than one database or CGI for example, the transaction can be broken up into discrete segments, data copied to and from the isolated storage system(s) or working systems as needed, and repairs can be run between each segment of a transaction, or between some segments of a transaction. Optionally, software can contain instructions that define what type of data can comprise a transaction, limiting the copy process to only copying data that meets certain criteria.

Cyber-Terrorism Examples

Cyber-terrorism represents a number of threats. One such threat occurs when e-mails are downloaded of which one e-mail contains a virus that when executed has the ability to infect other e-mail, infect the e-mail program so that it sends a copy of the virus with each new e-mail sent, and the virus places a hidden item in the operating system or applications that when executed after a period of two days, destroys the format or data structure or device drivers contained on any accessible data storage device, Such a virus may have been unknown and no protection or method of identification is available from virus-detection companies.

The protection process is described for processing e-mail, according to one embodiment. Upon download to the working system the unopened e-mails are then copied to the storage system (or alternatively they could be directly downloaded to the storage system) using a method in which the data cannot execute. A list of the e-mail subjects and who sent the e-mail and other pertinent information can be created and displayed to the user. For example this list could be generated by the storage system or the control system. User selects an e-mail to open. A copy of that e-mail is copied to the working system and then may be automatically opened. Optionally, a virus scan of the e-mail may be conducted. User reads and

responds to the e-mail, and the response may be copied to the storage system. A repair process may take place and repairs volatile or non-volatile data storage devices as needed.

Further, according to the example, a user selects next e-mail to open. This e-mail contains the virus. It is copied to the working system and is opened. No other e-mail is available for it to infect, but the e-mail infects the system folder used by that working system and several applications used in that working system. The user decides to respond to the e-mail and selects "respond". Optionally prior to responding, a repair process can be run or comparative process may be made between a master template and the working system. During the repair process or comparative process, the changes to the operating system associated with that working system or applications could be noted, and based on the difference(s) a virus warning could be drawn to the users attention, warning user not to respond to the e-mail as it may negatively affect the computer receiving the e-mail. Optionally a dialog can suggest that the user contact a virus alert center (ie. such as a national or international virus alert center that collects or responds to potential virus alerts.) and notify the center of the virus, or to allow the repair process to notify a virus alert center concerning the potential virus.

Optionally, based on certain criteria such as a virus threat analysis based on the type of changes made to the operating system or applications, the repair process could initiate commands to disable the network connection or e-mail software, or disable the e-mail process, or give the user a dialog indicating that based on the results of the virus threat analysis, the user may not be permitted to respond to the e-mail, and the ability to respond to that e-mail has been disabled. That e-mail could then be destroyed, or quarantined, or kept in isolation or kept in a storage system. Optionally such virus could be stored and deletion would not be permitted, pending approval from some entity, such as a virus alert center

that could authorize destruction of the virus by providing (for example) a code that would allow destruction of the virus. Optionally upon receiving such code the repair process could automatically destroy the virus laden e-mail. Optionally, the file could be encrypted or compressed, or modified in such a way that it could not execute and the repair process could send it to the virus alert center (with or without permission from the user.)

Optionally, such modification to computers and computing devices may be required by law, and the part of the repair process that dealt with potential viruses may be modified as needed to interact with government/commercial virus checking companies. For example a method of allowing upgrade of the software that dealt with viruses, permission to delete files, etc. may be required. In such cases specialized code could be created to interact with government agencies that would allow or require upgrade of the repair or virus checking software, allow or deny destruction of infected files, etc.

The repair process may run and make the working system identical to the master template, destroying all viruses, worms, and other changes in the process. The user finishes with the e-mail and selects the next e-mail. A repair may be conducted and then the next e-mail may then be copied to the working system, without risk of infection.

Loading a master template into volatile memory

In one embodiment, to further speed the repair process a master template of the working system and the software in the working system, may each be loaded into their own separate isolated volatile memory areas or shells to increase the speed of the repair process. Thus, if data in the working system is in volatile memory and the master template is in volatile memory, repairs can be conducted at higher speeds. Alternativly a new working system shell can be utilized, eliminating the need for a repair. For example a user could open an e-mail, and read the e-mail

using one shell, and if they want to respond to the e-mail a second shell could be used for the response. (Optionally the first shell can be checked for a virus while the user is writing a response to an e-mail using a second shell.) Additonal shells can be made ready for use.

In another embodiment, data can be downloaded directly to a storage system, using a method of encrypting or compressing or other copying which prevents execution of the data. A virus checking or repair process can be run as part of the repair sequence, or as a separate sequence. Optionally, an isolated hidden backup or archive system may be utilized with this invention, which may make an array of hidden backups or archives of the storage system or working system volatile or non-volatile memory/memories or data as desired, and which may be time stamped. Copying of data to such backup or archive system could also use techniques described herein to prevent execution of files and damage to the data on the backup system.

Optional information regarding copying or saving data

In one embodiment, the process of copying data may be dumb or restricted so that data being copied can't execute and thus the data on that data storage device can't be damaged by malicious code. For example, to move/copy data it can be encoded, or an ASIC can be utilized, or direct memory transfer or any other method of moving or copying data can be used that does not allow data to execute.

Optionally, copying could be orchestrated by a StoreExecute/control system that could have access to the isolated working system(s) and isolated storage system(s).

Selecting a file to open in the storage system could initiate a process whereby a file is copied from the storage system to the working system and opened. Saving a file in the working system could initiate a process whereby the file is copied to the storage system. Quitting a file in the

working system could initiate a process whereby the file is copied to the storage system and deleted in the storage system.

The term "copy" or "copies" or "copying" may be used in its broadest sense, and may include an algorithm, snapshot, compressed data, bit by bit, encryption, encoding, and the like.

Optional information explanation of data storage associated with the working system

Optionally, the data storage associated with a user working system could be temporary data storage, used while a file or files are needed or actively being worked on or needed by the system or the user. For example, when files were not being worked on they could be moved to the storage system, (ie. copied to the storage system and deleted from the working system). Thus, except for a copy of the Master Template located in the working system, data not being used is not stored on the working system data storage device where it would be potentially subject to being infected, damaged, destroyed, hacked, or manipulated in some way.

Optional use with web sites

Optionally, the working system could support a web site, or a computer could contain more than one working system or more than one storage system that could support various functions. For example one working system could contain a web site, while another working system is used by a user.

Optionally, one or more NetLock devices (described in the Appendices) may be used and may automatically switched or enable/disable network connections as desired.

Optionally, one or more NetLock devices may be used to switch, enable, or disable connections to a working system as needed.

Optionally, use of web software could indicate to a controller that is

associated with a Netlock Device and is process watching to enable a network connection to or from a working system, and quitting all network software (or lack of activity or other trigger) may indicate to a controller associated with the NetLock device to disable the network connection.

Optional explanation of automatic backup or archiving

Optionally, an automatic backup or archiving process may be associated with the storage system or the working system. Volatile or nonvolatile data may be saved, backed up or archived.

In one embodiment, external devices may be isolated and be used as storage systems. Alternatively, one or more external device(s) could also be isolated and used as one or more working systems. External ports can be connected to switches and switched, enabled, or disabled to connect to one or more isolated working systems, and then switched to connect to one or more isolated storage systems. Such switching may be done manually or automatically, or using a hardware switching process or a software switching process.

Optionally, in one embodiment, each time a save is made in a working system, a copy can be made to a storage system. Optionally, in order to prevent a virus or Trojan horse from causing havoc by performing millions of saves that get saved to the storage system, there could optionally be imposed a limit on frequency that a file could be saved, or other limitations could be placed on the process of saving data to the working system. (Optionally this could be part of the ROM or StoreExecute program.)

Optionally a quarantine data storage device can be used, or one or more common data storage device(s). Optionally, such data storage device can be accessed by the working system, or by the storage system, or by another logic control device that may also have access to the working system or storage system.

Optionally, a storage system may utilize one or more data storage devices. A working system can utilize one or more data storage devices. A working system and storage system can share a data storage device if they are isolated from each other. For example, a data storage device could be partitioned into two or more partitions, for example: Partition A and Partition B.

Optionally, working system "A" could consist of an isolated computing process associated with an isolated data storage partition located on partition "A". Storage system "B" could consist of an isolated computing process associated with an isolated data storage partition located on partition "B". Partitions can be isolated in a manner similar to how data storage devices can be isolated. Control over the partitions could optionally rely up an isolated computing process "C".

Optionally, applications and programs stored in the isolated working system can be repaired on command or automatically as needed. Optionally, a comparison process between a master template and the application/software in use could be used as a basis for how the application/software should look, and if different, components could be replaced as needed.

Optionally, a separate processor that has restricted functionality may be used to process data in the isolated working system, or the main processor can be given a restricted functionality. This can be done with multiple data storage devices, or one data storage device that has isolated partitions.

Optionally, the ability to execute files (located on a nonvolatile data storage device associated with a working system) may be enabled/disabled as needed. For example, logic control software may not contain code needed to execute files located on a nonvolatile data storage device associated with a storage system, or code needed to execute files can be disabled/enabled or switched on/off as needed.

Optionally, the logic control software associated with the storage system may be set to read only, or inaccessible from the working system or storage system (so that malicious code can not effect the software nor the processor nor gain access to the storage system). Optionally, a third isolated logic control and computing processes may be used to access that code. A logic control and computing processes may be performed via separate logic control and processing devices, or be on a single device that has the ability to isolate two or more logic control processes.

Optionally, data that is copied from the volatile or nonvolatile data storage device(s) associated with the working system to the storage system can be deleted from the working system and associated data storage devices as needed. This may help to prevent hacking, etc.

Optionally, working system(s) or their associated Data Storage Devices, and storage system(s) or their associated Data Storage Devices, need not be on a computing device together. They can be on a network, external, have wireless connections, or be anywhere. For example, a computing device may have a working system, in which an associated nonvolatile data storage device is in a nearby server; and a storage system may be located over a network, and associated with an external wireless data storage device.

Optionally, a working system may not have an associated non-volatile data storage device. A working system could be limited to volatile storage. Additionally, a working system may have a plurality of processing functions or processors associated with it.

In one embodiment a switching process that may be controlled by the control system that may be used to switch which system(s) have access to network communication. Network communication can be dedicated to a particular working system/ or storage system, or switched as needed.

Optional Shells

Optionally, using a variation of the Shell approach, isolated shells may operate as working systems optionally with associated data storage, and other isolated shells can operate as storage systems optionally with associated data storage. Data may be copied to and from to the working system and storage system shells associated volatile or nonvolatile memory using a copy process that prevents the execution of data.

Optional changes to software

Optionally, in order to enhance the effectiveness of the isolated working system & storage system embodiments described above, the following changes may be made to software. Data used by the software may be kept in a storage system until needed. Data can be broken up and only data pulled from the working system that is needed. For example, instead of treating an e-mail in box as one file, e-mail programs can be modified to treat them as separate files, and only copy specific file(s) into or out of the working system as needed, keeping all of the other data isolated. Alternatively, data could be stored in the working system as one or more files, but when for example a specific e-mail was needed, only that specific e-mail part of a file could be copied to the working system, and data could be saved from the working system into that one file in the storage system.

E-mail was used here as an example. Optionally, software, and especially software used for the web, may use the approach of storing records as individual files, or keeping them in one or more files and only bringing the data into the working system that is needed at that time or is likely to be needed.

Optionally, when a Netlock device enables internet connection e-mail and other software used on the web that is currently in the working system may be limited to only data that needs to be sent or used, limiting a hackers ability to access any other data. During web commerce sessions,

data can be frequently moved to and from the storage system as needed to ensure that the least possible, preferably only that data required and in use or needed for use is in the working system.

Optionally, an index or database containing content of some data or files contained in the storage system may be moved to or located in the working system. When such data is selected to use or open, it could then be copied into the working system as needed and copied back to the storage system when not needed, and deleted from the working system.

Optionally, switching data storage device identity may be done using software that interacts with the data storage device or data storage device controller. Such software could be isolated from the working system and storage system. For example it could be part of an isolated StoreExecute that conducts the repair process, or it could be on it's own isolated StoreExecute. This may necessitate a change in some data storage device controllers to enable them to accept software commands to change identity/boot sequence.

Optionally, a data storage device may be hot swappable, and turned on only as necessary during the isolated backup event.

Optional Netlock

Optionally, the netlock device may be controlled by any type of logic control device, triggered automatically or manually, by a hardware or software process. Switch trigger may include or utilize a timer/scheduler. It may also include any method of triggering a switching process. For example, a coin operated mechanism or pin card operated mechanism could be used that triggers netlock. A dual or multi-line version of netlock that can deal with more than one network connection (two or more network connections), in which case the netlock device may optionally be modular in nature to add additional network connections as needed.

Optionally, a dual or multi -line version of netlock that can deal with more than one network connection (two or more network connections), in which case the netlock device may optionally be modular in nature to add additional network connections as needed. If so desired the multi-line version could potentially controlled by one logic controller or switching process.

The inventions and methods described herein can be viewed as a whole, or as a number of separate inventions that can be used independently or mixed and matched as desired. All inventions, steps, processed, devices, and methods described herein can be mixed and matched as desired. All previously described features, functions, or inventions described herein or by reference may be mixed and matched as desired.

Optionally, a process hereinafter referred to as an Installer Watcher, may run in the background of a computer that can look for activity that appears to be an installer. If the user attempts to install software, the attempt at installation may be halted and a dialog could query the user as to whether the user is installing software. If so the Installer Watcher could walk the user through a process of installation or testing the software prior to updating a Master Template or during actual update of a Master Template.

The foregoing descriptions of specific embodiments and best mode of the present invention have been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed, and obviously many modifications and variations are possible in light of the above teaching. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, to thereby enable others skilled in the art to best utilize the invention and various

embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the claims appended hereto and their equivalents.

5

WHAT IS CLAIMED IS:

1. A method for a computer repairing itself, the method comprising the computer-executed steps of:

booting from a first boot device;

then, in response to a signal indicating a need for repair, booting from a second boot device; and

then repairing software on the first boot device while booted from the second boot device.

2. The method of claim 1, wherein the step of repairing software comprises

copying software from a device other than the first boot device onto the first boot device.

3. The method of claim 1, wherein the step of repairing software comprises

copying software from the second boot device onto the first boot device.

4. The method of claim 1, wherein the step of repairing software comprises

copying template, backup and/or archive software from a device other than the first boot device onto the first boot device.

5. A method for a computer repairing itself, the method comprising the computer-executed steps of:

booting from a first boot device;

then, in response to a signal indicating a need for repair, booting from a second boot device; and

then, while booted from the second boot device, copying template, backup and/or archive software from the second boot device onto the first boot device.

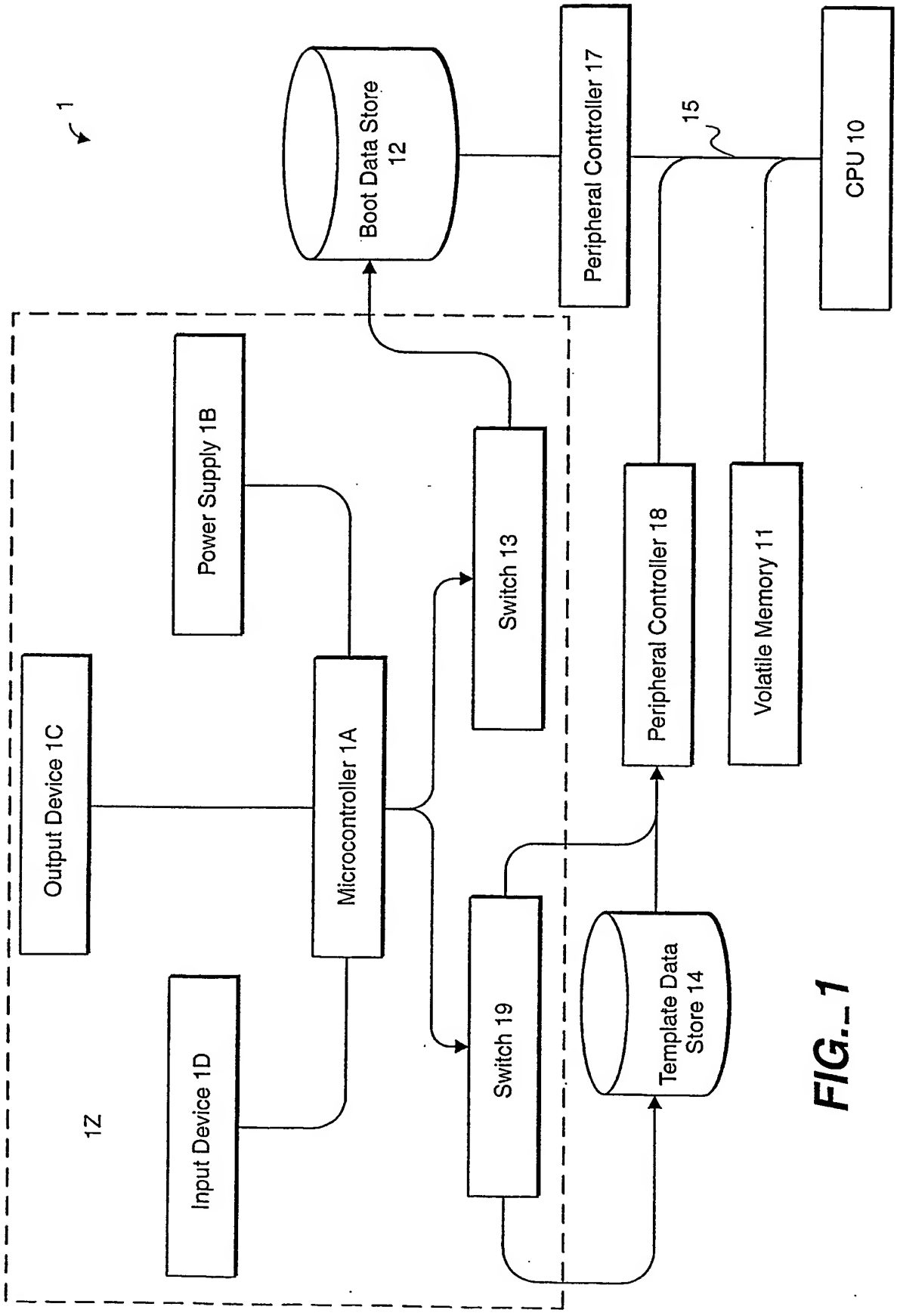

1
Boot Data Store 12
Peripheral Controller 17
15
CPU 10
Power Supply 1B
Switch 13
Peripheral Controller 18
Volatile Memory 11
Output Device 1C
Microcontroller 1A
Switch 19
Template Data Store 14
Input Device 1D
1Z

FIG._1

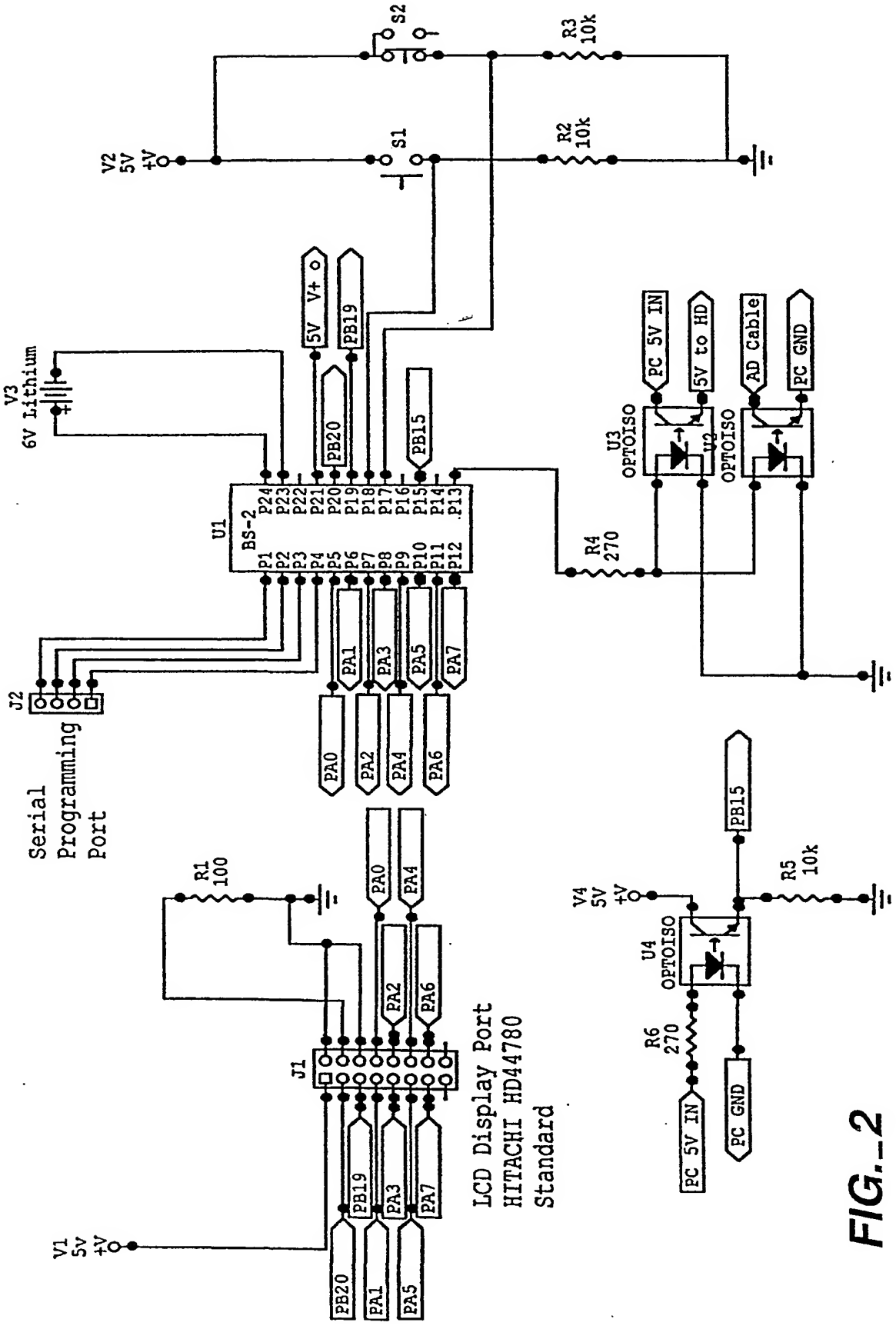

S2
R3
10k
S1
R2
10k
V2
5V
+V
5V V+
PB19
PB20
PB15
V3
6V Lithium
U1
BS-2
P24 P23 P22 P21 P20 P19 P18 P17 P16 P15 P14 P13
P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12
PA1 PA3 PA5 PA7
PA0 PA2 PA4 PA6
J2
Serial
Programming
Port
PC 5V IN
5V to HD
AD Cable
PC GND
U3
OPTOISO
U2
OPTOISO
R4
270
R1
100
PA0 PA4
PA2 PA6
J1
LCD Display Port
HITACHI HD44780
Standard
V1
5V
+V
PB20 PB19 PA1 PA3 PA5 PA7
V4
5V
+V
PB15
R5
10k
U4
OPTOISO
R6
270
PC 5V IN
PC GND

FIG._2

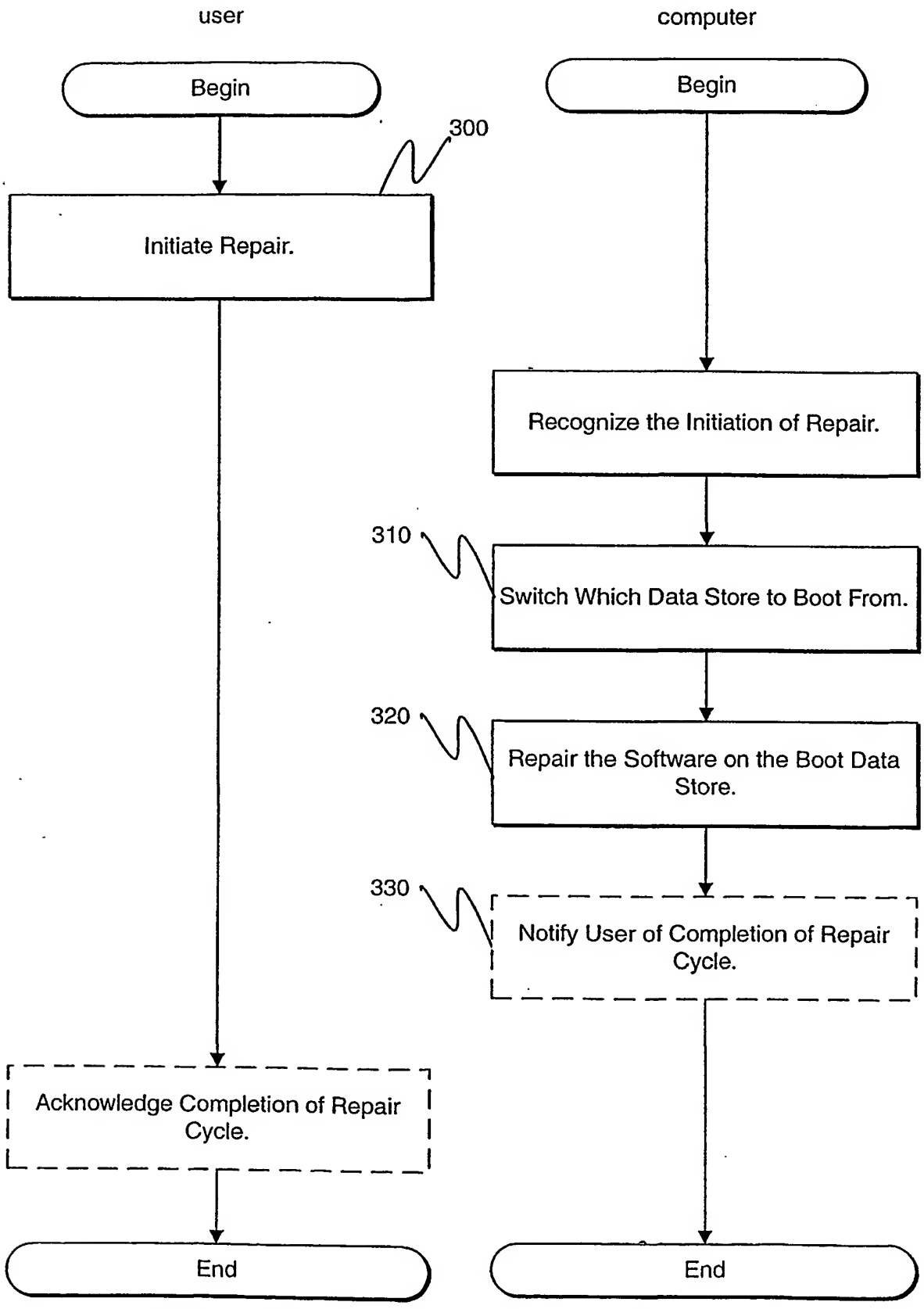

user
Begin
300
Initiate Repair.
Acknowledge Completion of Repair Cycle.
End
computer
Begin
Recognize the Initiation of Repair.
310
Switch Which Data Store to Boot From.
320
Repair the Software on the Boot Data Store.
330
Notify User of Completion of Repair Cycle.
End

FIG._3A

FIG._3B

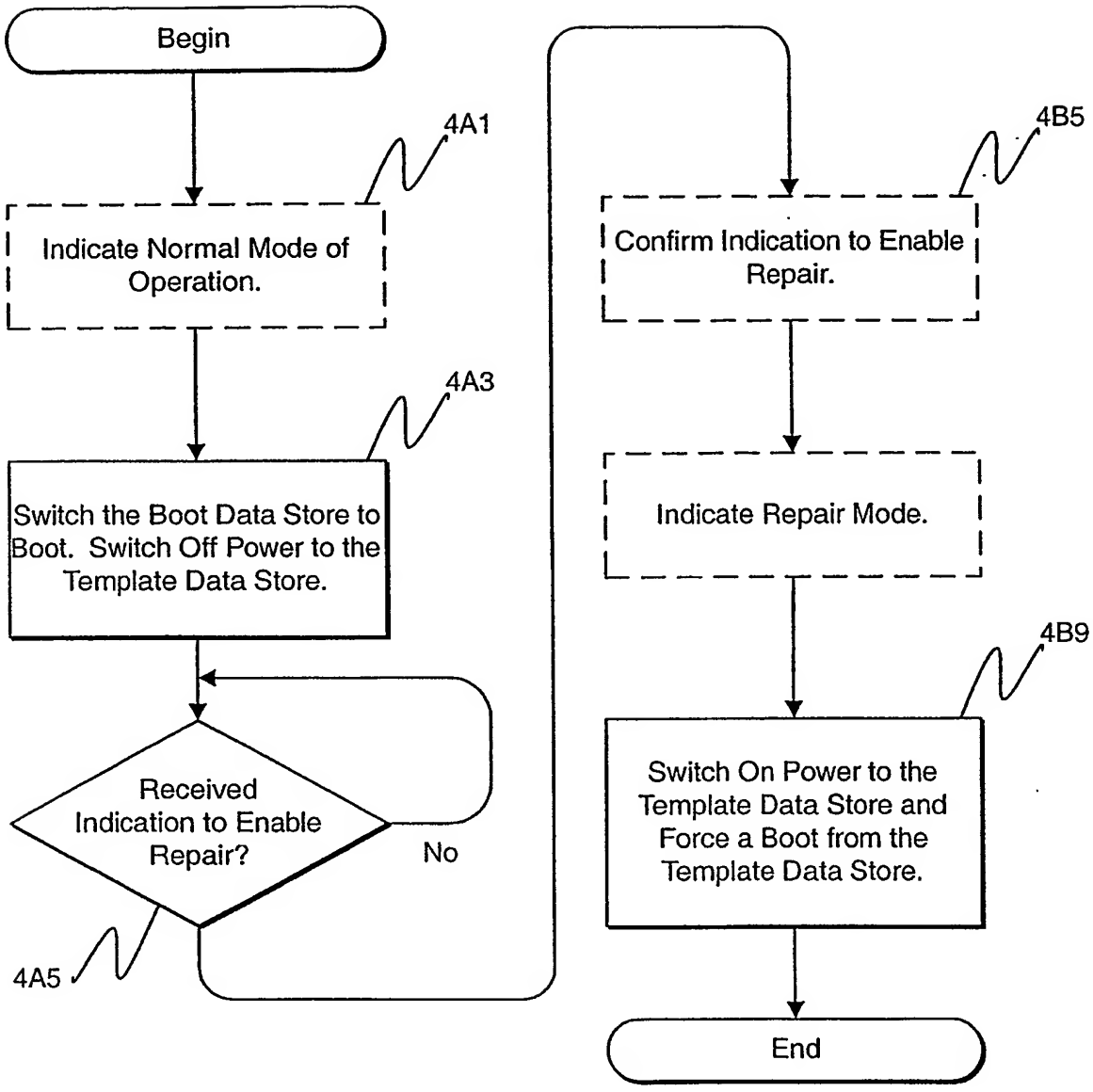

Begin
4A1
Indicate Normal Mode of Operation.
4A3
Switch the Boot Data Store to Boot. Switch Off Power to the Template Data Store.
Received Indication to Enable Repair?
No
4A5
4B5
Confirm Indication to Enable Repair.
Indicate Repair Mode.
4B9
Switch On Power to the Template Data Store and Force a Boot from the Template Data Store.
End

FIG._4

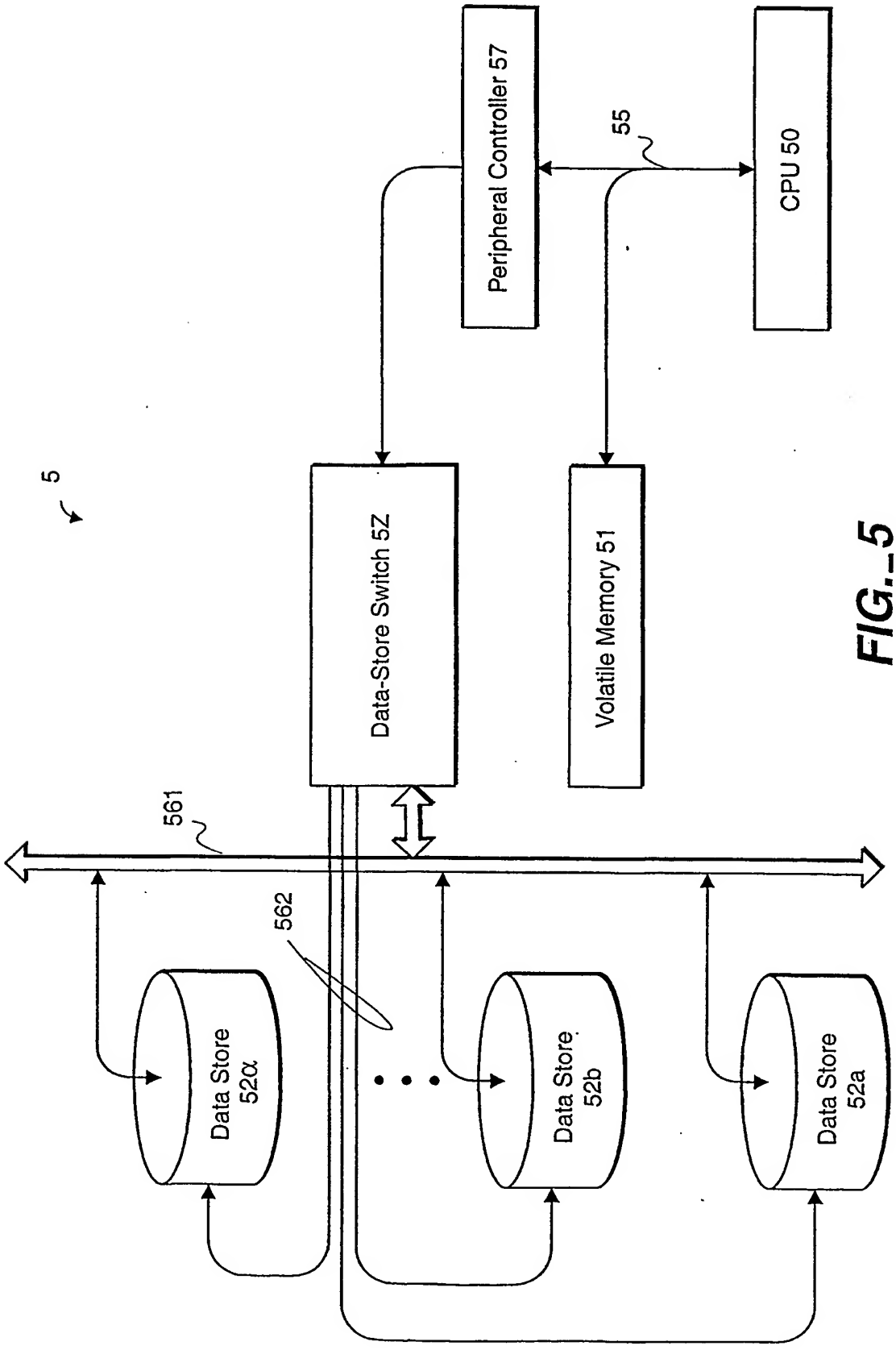

Peripheral Controller 57
55
CPU 50
5
Data-Store Switch 5Z
Volatile Memory 51
561
562
Data Store 52α
Data Store 52b
Data Store 52a

FIG._5

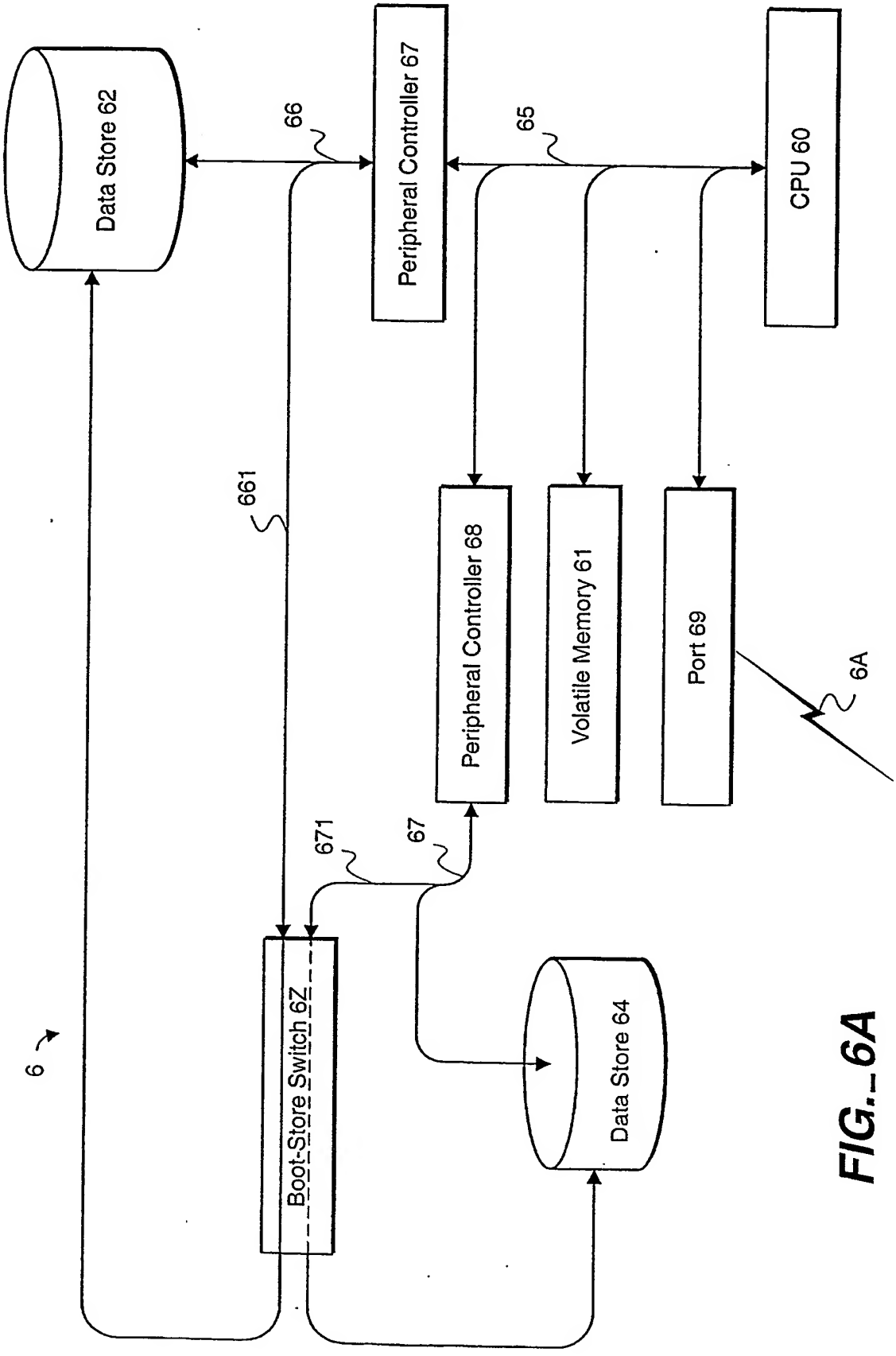

Data Store 62
66
Peripheral Controller 67
65
CPU 60
661
Peripheral Controller 68
Volatile Memory 61
Port 69
6A
671
67
6
Boot-Store Switch 6Z
Data Store 64

FIG._6A

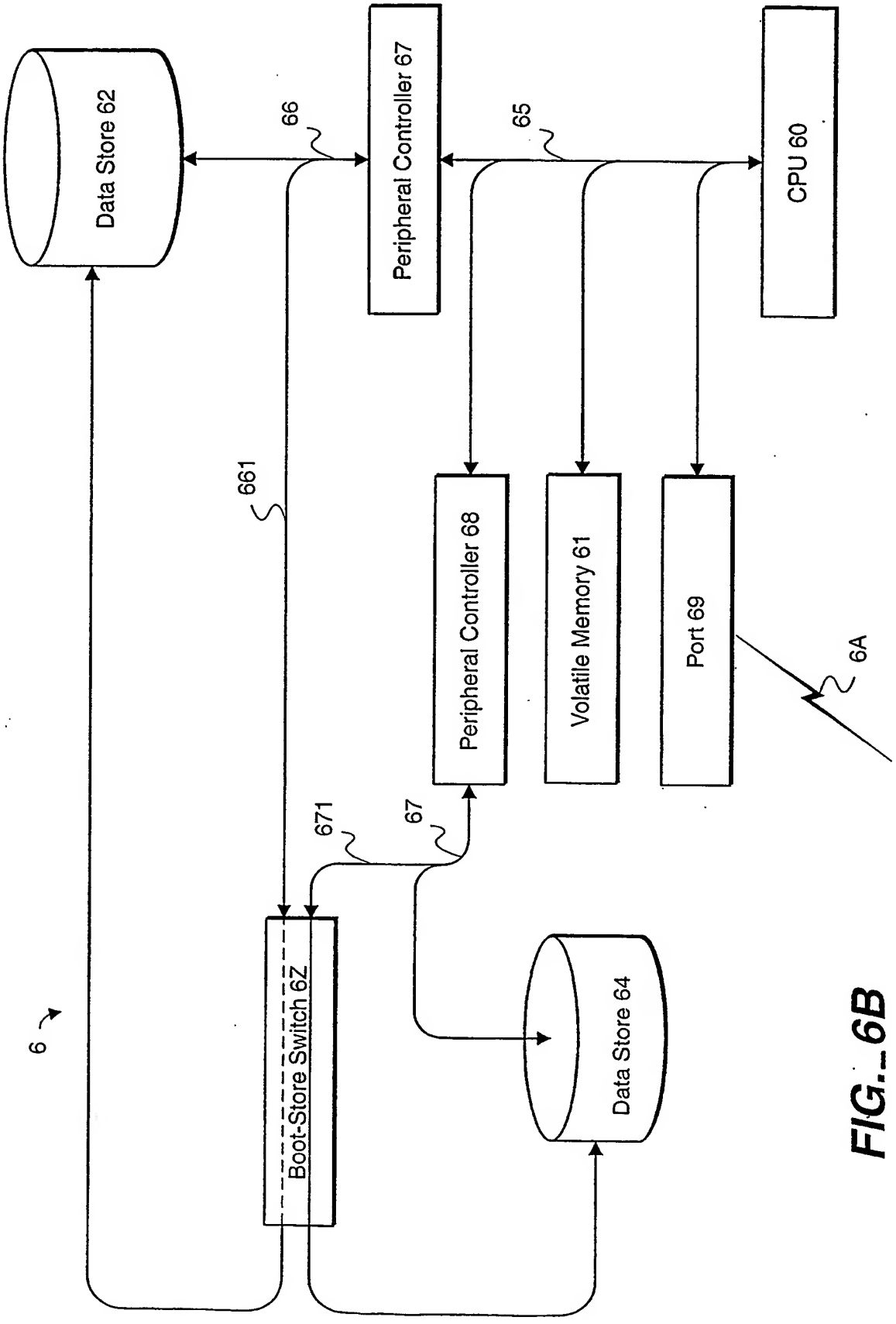

Data Store 62
66
Peripheral Controller 67
65
CPU 60
661
Peripheral Controller 68
Volatile Memory 61
Port 69
6A
671
67
Boot-Store Switch 6Z
Data Store 64
6

FIG._6B

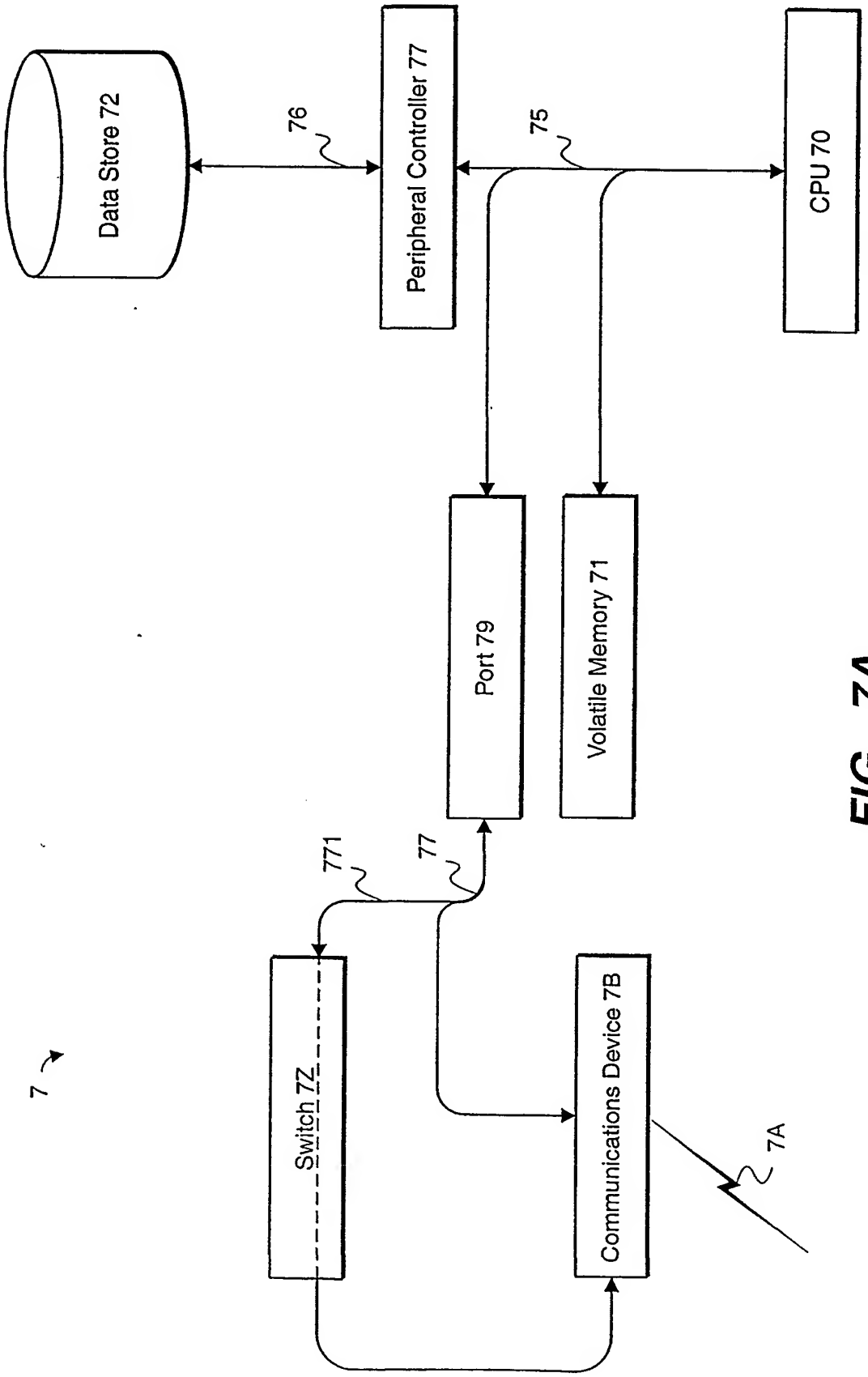

Data Store 72
76
Peripheral Controller 77
75
CPU 70
Port 79
Volatile Memory 71
771
77
7
Switch 77
Communications Device 7B
7A

FIG._7A

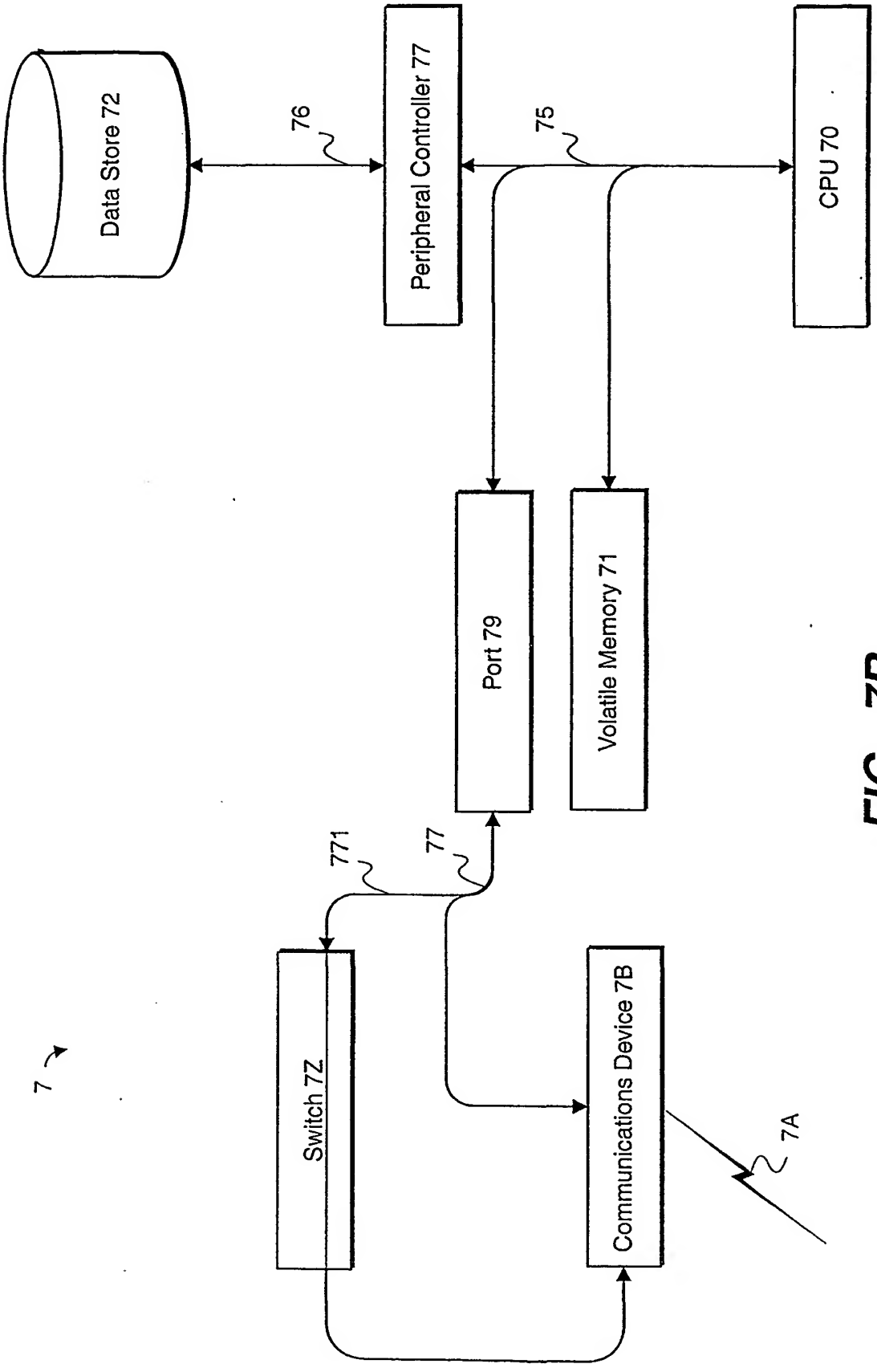
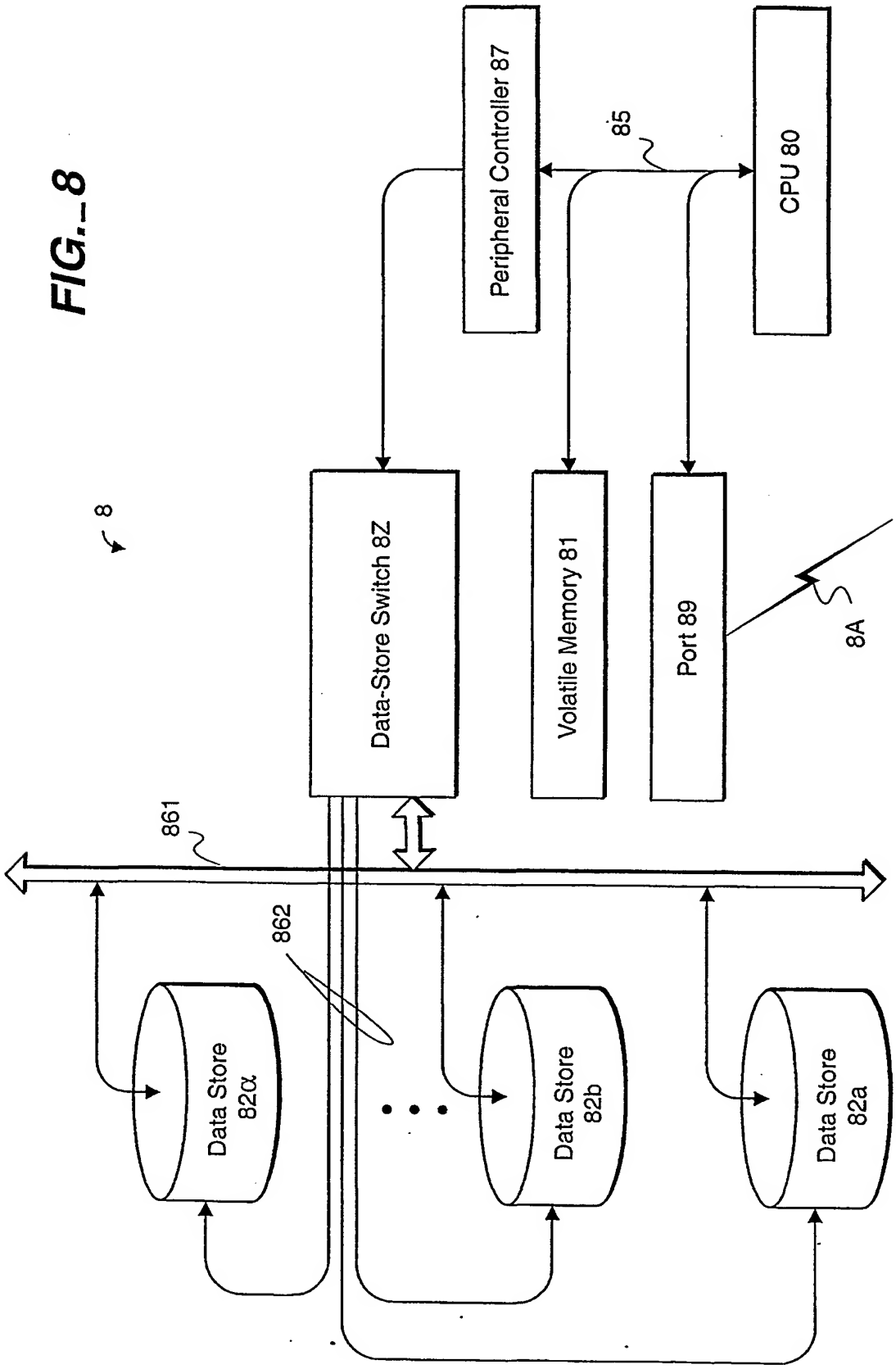

Data Store 72
76
Peripheral Controller 77
75
CPU 70
Port 79
Volatile Memory 71
771
77
7
Switch 7Z
Communications Device 7B
7A

FIG._7B

FIG._8


8
Peripheral Controller 87
85
CPU 80
Data-Store Switch 8Z
Volatile Memory 81
Port 89
8A
861
862
Data Store 82α
Data Store 82b
Data Store 82a

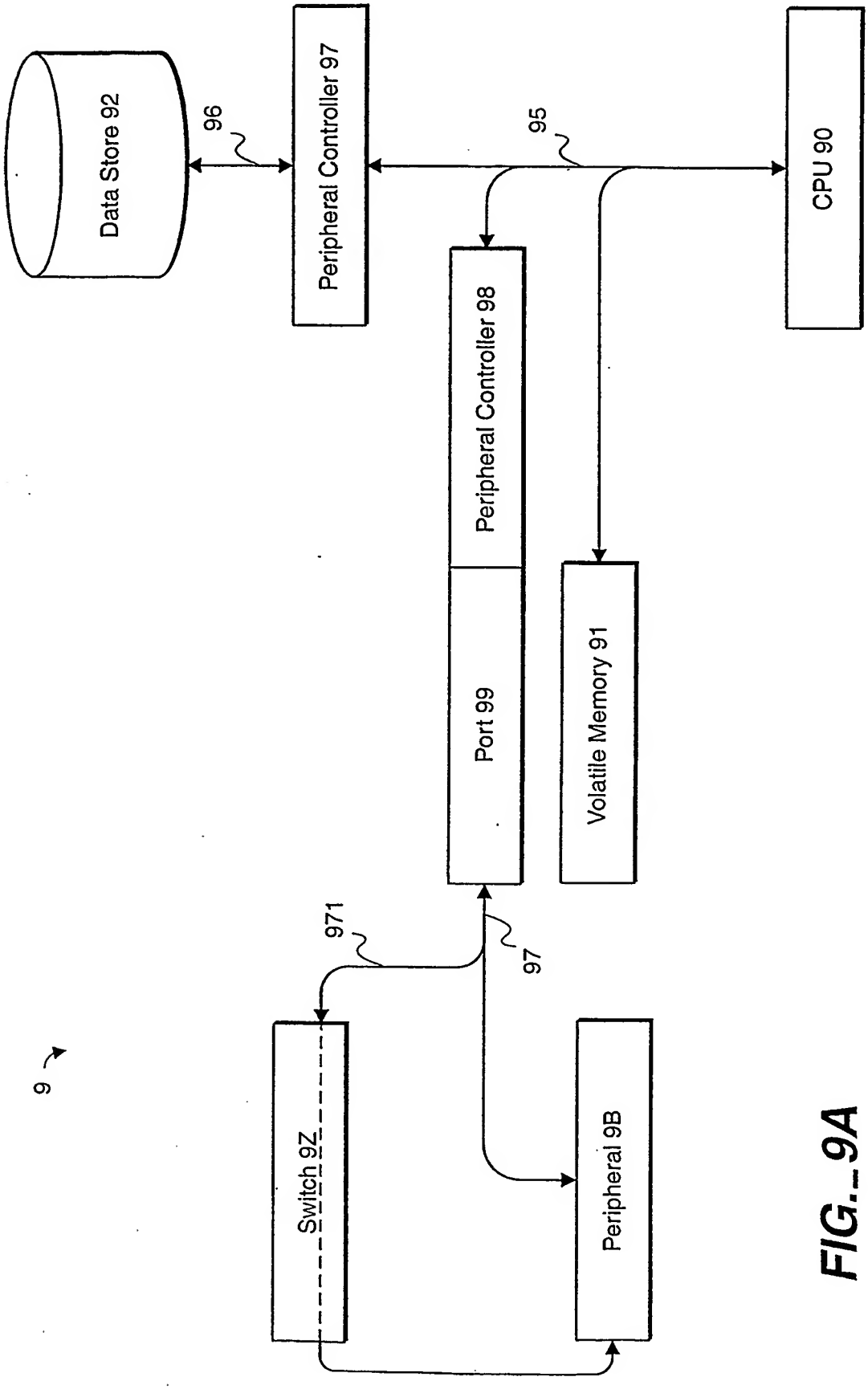

Data Store 92
96
Peripheral Controller 97
95
CPU 90
Peripheral Controller 98
Port 99
Volatile Memory 91
971
97
9
Switch 9Z
Peripheral 9B

FIG._9A

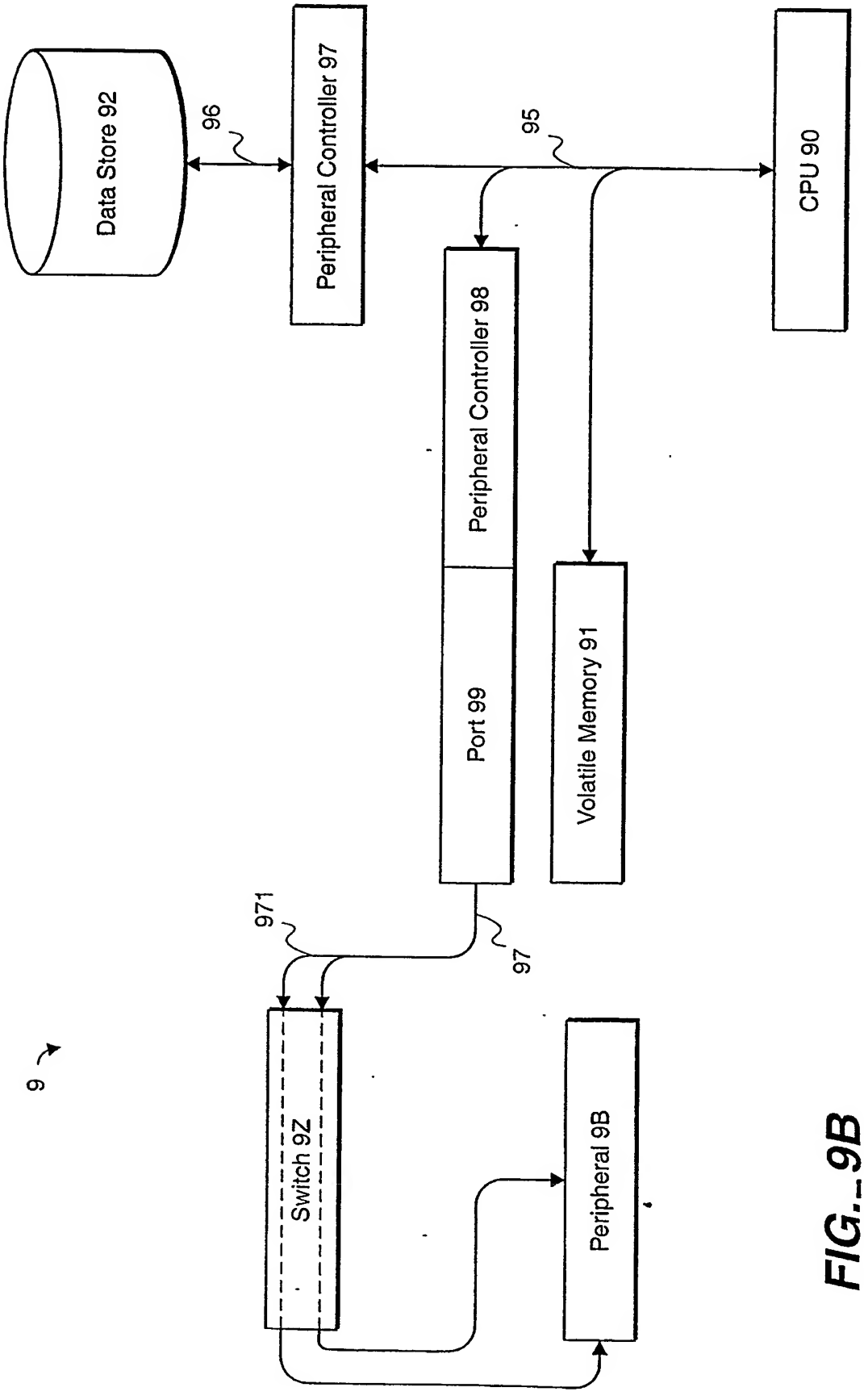

Data Store 92
96
Peripheral Controller 97
95
CPU 90
Peripheral Controller 98
Port 99
Volatile Memory 91
971
97
Switch 9Z
Peripheral 9B
9

FIG._9B

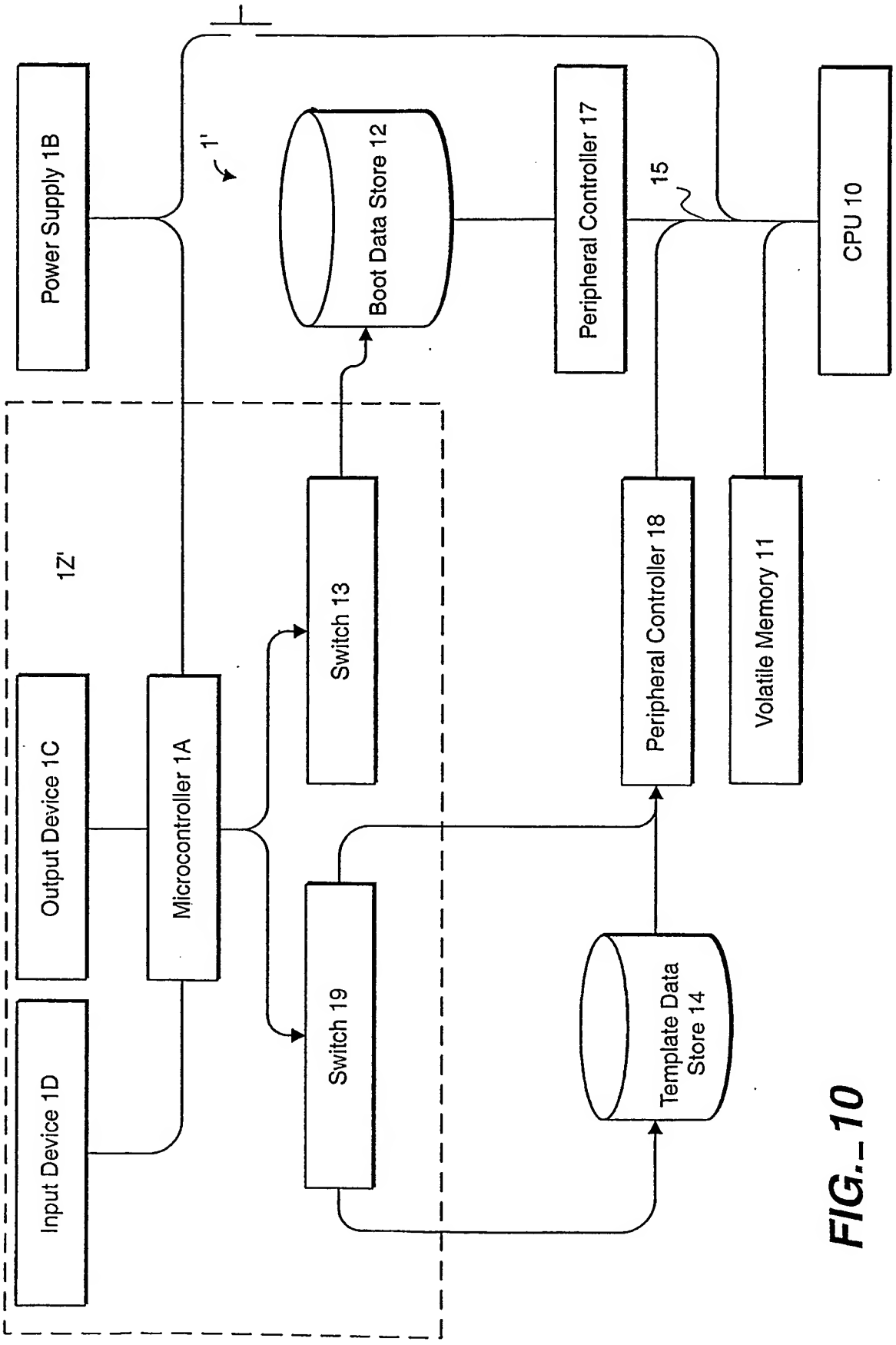

Power Supply 1B
1'
Boot Data Store 12
Peripheral Controller 17
15
CPU 10
1Z'
Switch 13
Peripheral Controller 18
Volatile Memory 11
Output Device 1C
Microcontroller 1A
Switch 19
Template Data Store 14
Input Device 1D

FIG._10

Description of Self-Repairing System:

When power is applied to the system, the micro controller powers up, initializes the LCD prompt screen and the "Ready for Normal Operation" prompt is displayed. At this point the micro controller waits until the single pole single throw switch (S1) is pressed. When S1 is pressed the micro controller prompts the user "Repair Selected".

The position of the slider switch S2 is now stored in memory. Then based on the switch's initial location the user is instructed to slide the switch opposite its initial position to confirm the repair request: "Please slide switch from A->B" or "Please slide switch from B->A".

At this point or any other time (except when a "Repair in Progress"), the user can cancel the repair request by pressing the pushbutton switch, at which point the dialog "Repair Cancelled" is displayed. The system will then return to "Ready for Normal Operation" mode.

If the user wishes to confirm the repair, and moves the slide switch as instructed, the computer will check the power status of the computer. The micro controller reads the logic level of opto-isolator U4. Logic high represents the computer being powered. If the computer is on, the user will be prompted to "Please Shutdown the Computer". The computer then again reads in the logic level of U4.

If the computer has been shut down U4 will output a logic low. After the controller confirms that the computer is off the user is then prompted to restart the computer

At this point solid-state relay U3 and opto-isolator U2 are enabled. This ensures that when the computer is restarted the Master hard drive receives power as IDE ID0 and the internal user drive boots with its ID switched to Address 1 (ID1).

When the computer is restarted, U4 outputs logic high to the micro controller. This signal denotes that the computer is powered and prompts the "Repair in Progress" message, which is displayed on the LCD screen. Once the repair process is enabled the micro controller displays this message and executes no other commands until the computer is shutdown (U4 outputs a logic low signal).

When the computer is shutdown by the repair software, the micro controller informs the user that the process is complete with the "Repair Complete" dialog.

The micro controller then enters the "Ready for Normal Operation" mode and waits for a new repair process.

Hardware description:

The microcontroller at the heart of the system is a Basic Stamp II micro controller which controls the repair progress, LCD user interface and also supplies a 5 volt regulated power supply for all the devices in the circuit.

FIG._11A

The optional LCD screen is a 2X16 character LCD based on the industry standard Hitachi HD44780 controller chip.

The solid-state relay, which switches power to the master hard drive, is an International Rectifier PVN012. Any switching device could be used to switch hard drive power.

The other opto-isolators are NEC 2501.

The batteries are quantity 2 – 3 volt 1200ma Lithium batteries connected in series for a 6 volt output. This powers a voltage regulator on the Micro controller, which supplies 5 volts to the rest of the circuit. Any power supply from 5.1 volts to 15 volts can be used.

FIG._11B

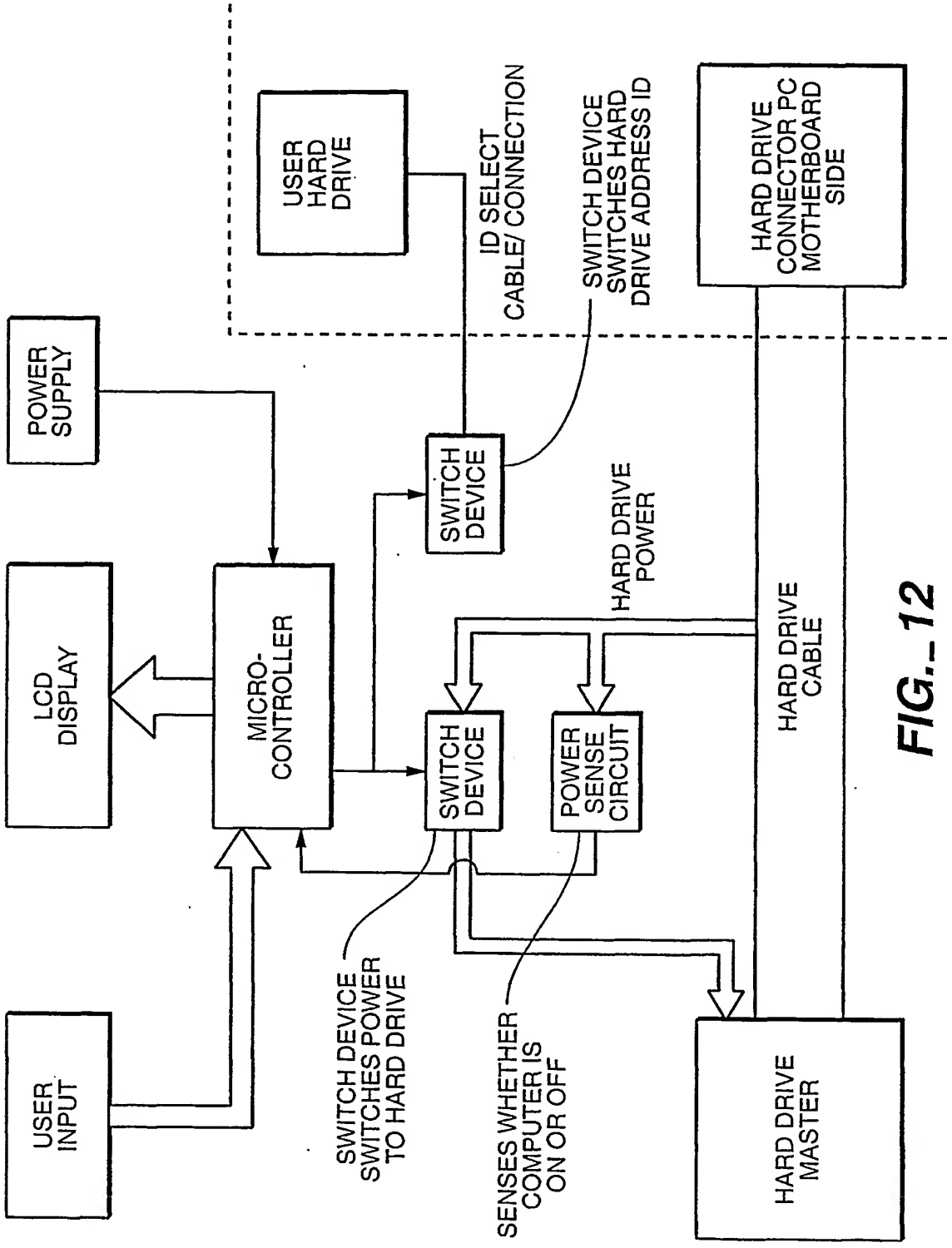

USER HARD DRIVE
ID SELECT CABLE/ CONNECTION
SWITCH DEVICE SWITCHES HARD DRIVE ADDRESS ID
HARD DRIVE CONNECTOR PC MOTHERBOARD SIDE
POWER SUPPLY
SWITCH DEVICE
HARD DRIVE POWER
LCD DISPLAY
MICRO-CONTROLLER
SWITCH DEVICE
POWER SENSE CIRCUIT
HARD DRIVE CABLE
USER INPUT
SWITCH DEVICE SWITCHES POWER TO HARD DRIVE
SENSES WHETHER COMPUTER IS ON OR OFF
HARD DRIVE MASTER

FIG._12

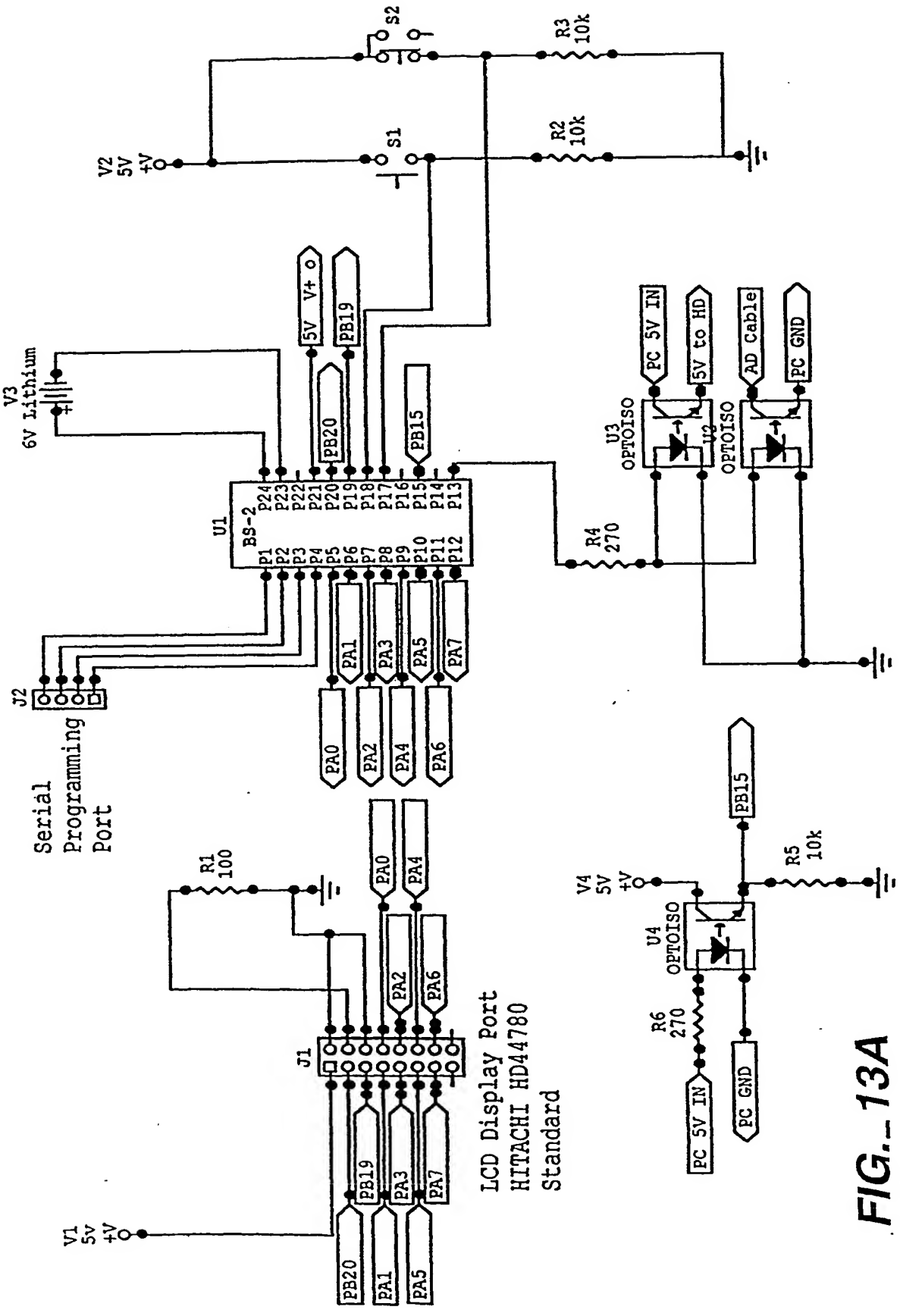

S2
R3
10k
S1
R2
10k
V2
5V
+V
V3
6V Lithium
5V V+
PB19
PB20
PB15
U1
BS-2
P24 P23 P22 P21 P20 P19 P18 P17 P16 P15 P14 P13
P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12
PA1 PA3 PA5 PA7
PA0 PA2 PA4 PA6
R4
270
U3
OPTOISO
U2
OPTOISO
PC 5V IN
5V to HD
AD Cable
PC GND
J2
Serial
Programming
Port
R1
100
PA0 PA4
PA2 PA6
J1
LCD Display Port
HITACHI HD44780
Standard
V1
5v
+V
PB19 PA3 PA7
PB20 PA1 PA5
V4
5V
+V
PB15
R5
10k
U4
OPTOISO
R6
270
PC 5V IN
PC GND

FIG._13A

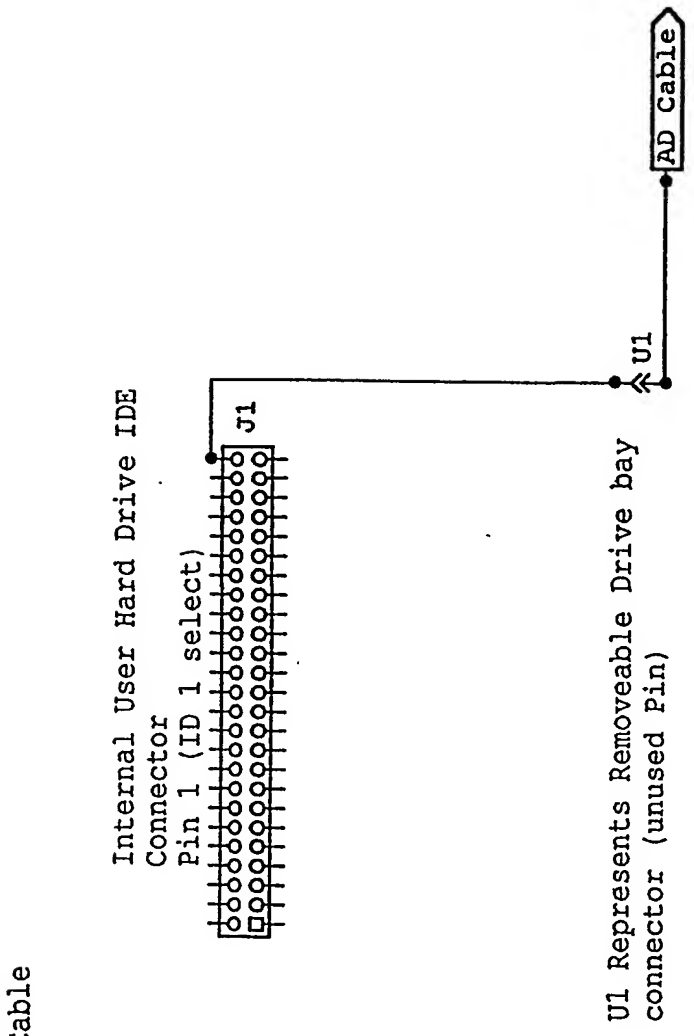

Address Select Cable
Internal User Hard Drive IDE
Connector
Pin 1 (ID 1 select)
J1
U1
AD Cable
U1 Represents Removeable Drive bay
connector (unused Pin)

FIG._13B

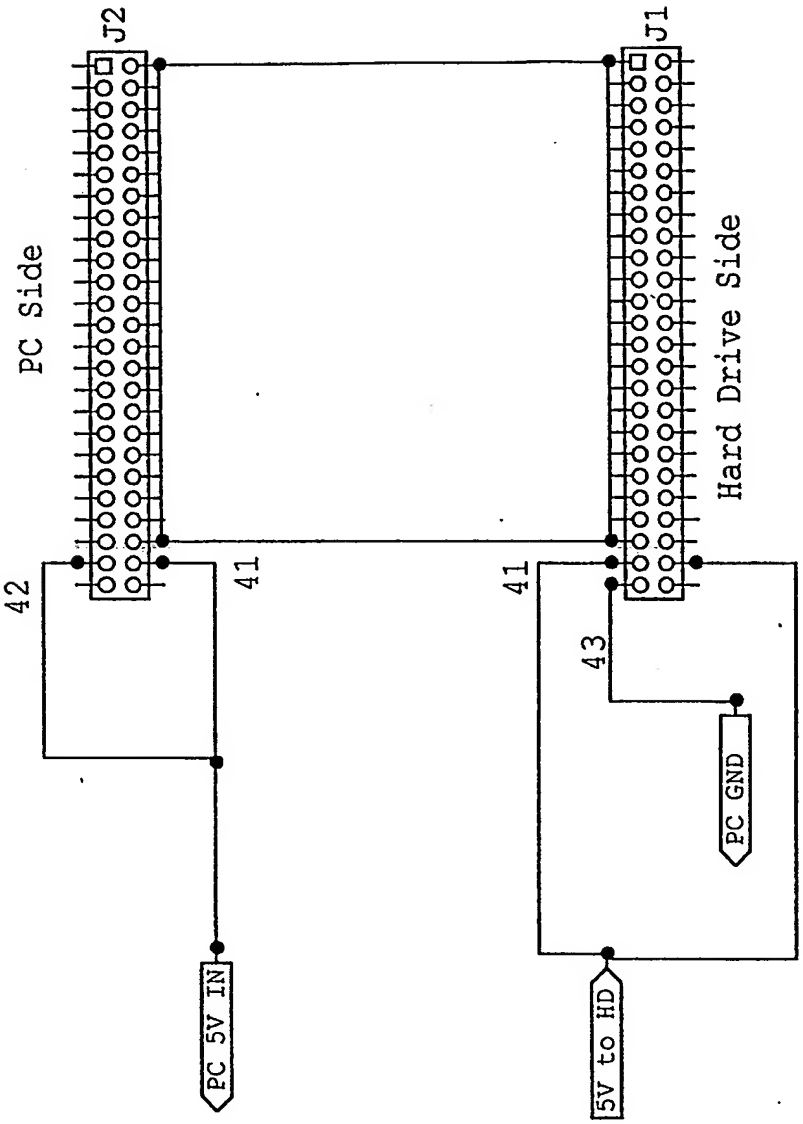
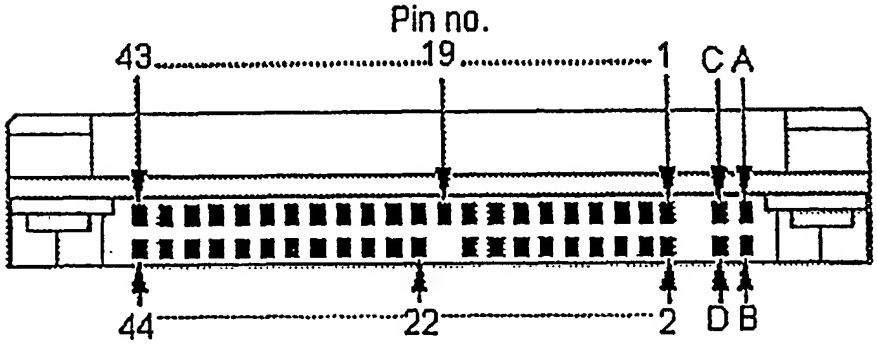

J2
J1
PC Side
Hard Drive Side
42
41
41
43
42
PC 5V IN
5V to HD
PC GND

FIG._13C

Cable Pinout

Hard Drive Pinout (from IBM website)

Travelstar 20GN, 30GT, 32GH
Models: DJSA-232, 230, 220, 210, 205



Notes: A/T = IDE. Pin position 20 is left blank for secure connector insertion. Pin position A,B,C, and D are used for drive address setting.
Signal definition

As shown in the schematic diagrams, the IDE Hard Drive cable is a standard 'Hard Drive extension cable. The Hard Drive power connections (cable lines 41 and 42 have been cut and route through the circuit's solid state relay which switches the hard drive power under the microcontroller's direction. PC ground connection (line 43) has also been routed to the circuit board to drive the opto-isolators. The computer power supply and the microcontroller power supply are separate, and interfaced through opto-isolators. Therefore the microcontroller ground and PC ground must be kept separate

The Address Selector Cable carries the address signal connection from the internal User Hard drive to the removeable drive bay, removeable connector. On that connector an unused pin is connected on the computer mother board side. The male interface connector from the removeable drive bay then interfaces that signal when the bay is inserted into the computer. On the drive bay side the signal is then taken to the micrcontroller where the connection is switched to ground in order to change the drive address (ID select signal pin is a floating pin that must be sinked to ground in order switch).

In other systems with different types of hard drives there are other methods of accomplishing the switching process. The necessary signals to switch are the Hard Drive power and Drive ID. These connections may different on other computer hard drives. For example ID switching could be carried out using Cable Selector. The Computer power sensing circuit could also sense power somewhere else in the computer.

FIG._14

```
'.....
'       Variable Declarations


Sw1                     var             IN13
Sw2                     var             IN12
Sw2_Loc                 var             byte
Sw2_Loc_Cnfrm   var             byte
Mode                    var             byte
Char                    var             b3
btnWk           var             byte
HD                      var             Out8
Comp_On         var             in10
z                       var             byte
RS                      con             15
E                       con             14
LINE1           con             128
LINE2           con             128+$40
'.....
'       Variable initializations

DIRs = %1100000011111111
btnWk=0
'.....

Init_LCD:                                       'LCD INITIALIZATION
low 8                                           'Hard Drive OFF


low RS
low E
x var byte
x=0

i_LCD:  OUTL = %00000010                        'Set to 8-bit operation.
                        pulsout E, 1                            'Send data three times
                pause 20                        'to initialize LCD.
                        pulsout E, 1
                pause 20
                        pulsout E, 1
                pause 20

                        outl = $0038            'Set up LCD in accordance with
                pulsout E,1             'Hitachi instruction manual.

               outl = %00001010         'Turn on cursor and enable
                pulsout E,1             'left-to-right printing.

                        outl = %00001100
                         pulsout E,1

                        outl = %00000001
                         pulsout E,1

                        outl = %00000110
                         pulsout E,1

        High RS                                 'Prepare to send characters.
                        outl = %00000101
                         pulsout E,1
                        outl = %00000011
                         pulsout E,1

        Gosub CLR
```

FIG._15A

```
.....
Start:                                                  'idle mode
      HD=0                                                    'Hard Drive off - Normal operation
      gosub Ready_Mesg

Idle:
      BUTTON 13,1,255,0,btnWk,1,Repair                  ' check pushbutton switch

      goto Idle


Repair:
            gosub GETSW_LOC

            if SW2_LOC= "A" then UPDLOC
                  SW2_LOC_CNFRM= "A"

            goto Repair1

            UPDLOC:
            SW2_LOC_CNFRM= "B"

Repair1:

      gosub CLR

            low RS
            x=LINE1+4                                   'center text on top line
            gosub Send_data
            high RS


      for b2=0 to 14
            Lookup b2,["Repair Selected"],b3
            x=b3
            gosub Send_data
            next
      pause 500


            low RS                                                  'format text
            x=Line2+1
            gosub send_data
            high RS


      for b2=0 to 17
            Lookup b2,["Slide Switch From "],b3
            x=b3
            gosub Send_data
      next


      x=SW2_Loc
      gosub send_data

      x="-"
      gosub send_data

      x=">"
      gosub send_data

      x=SW2_Loc_Cnfrm
      gosub send_data
```

FIG._15B

```
Cnfrm_loop:
            gosub GetSW_Loc                                    ' Get Current Switch Location
            BUTTON 13,1,255,0,btnWk,1,Cancel                   ' check pushbutton switch

            if SW2_Loc = SW2_Loc_Cnfrm Then Enable_Repair      'check for confirmation or cancellation

goto Cnfrm_loop


Enable_Repair

      if Comp_On = 1 then Prompt_Shutdown

      goto Prompt_Restart


Prompt_Shutdown:

      gosub Shutdown_Mesg


Shutdown_Loop:

      If Comp_On = 0 then Prompt_Restart
      BUTTON 13,1,255,0,btnWk,1,Cancel

goto Shutdown_Loop



Prompt_Restart

      HD = 1                                    'Enable Hard Drive - Normal Addressing
      gosub Restart_Mesg

Restart_Loop:

      If Comp_On=1 Then In_Progress

      BUTTON 13,1,255,0,btnWk,1,Cancel
goto Restart_Loop

      Gosub Progress_Mesg
      pause 200
      If Comp_On=0 then Repair_End
      pause 300
      gosub ClR
      Pause 200
      If Comp_On=0 then Repair_End
      Pause 300
      Gosub ClR
Goto In_Progress

Cancel:
      gosub CLR
      pause 400
      gosub Cancel_Mesg
      pause 400
      gosub CLR
      pause 400
      gosub Cancel_Mesg
      pause 400
      gosub CLR
      pause 400
      gosub Cancel_Mesg
      pause 400
```

FIG._15C

```
goto Start
Repair_End:

        HD = 0                          'Disable Hard Drive - Return to Normal Mode

        Gosub CLR
        pause 400
        Gosub Complete_Mesg
        pause 400
        Gosub CLR
        Gosub Complete_Mesg
        pause 400
        Gosub CLR
        Gosub Complete_Mesg
        pause 400
        Gosub CLR

goto Start

'''''''
```

```
return
'*************************************************************************
Cancel_Mesg:

        gosub CLR


        Low RS
        x=LINE1+4                       'center text on top line
        gosub Send_data
        high RS

    for b2=0 to 15
        Lookup b2,["R","e","p","a","i","r"," ","C","a","n","c","e","l","l","e","d"],b3
        x=b3
        gosub Send_data
        next

return
'*************************************************************************
Shutdown_Mesg:

    Gosub CLR

        for b2=0 to 23
        Lookup b2,["P","l","e","a","s","e"," ","s","h","u","t","d","o","w","n","
","C","o","m","p","u","t","e","r"],b3
        x= b3
        gosub send_data
        next
return
'*************************************************************************
Restart_Mesg:

    Gosub CLR

        for b2=0 to 22
        Lookup b2,["P","l","e","a","s","e"," ","R","e","s","t","a","r","t","
","C","o","m","p","u","t","e","r"],b3
        x= b3
        gosub send_data
        next
return
'*************************************************************************
Progress_Mesg:

    gosub CLR

        low RS
        X=LINE1+3                       'center text on top line
        gosub Send_data
        high RS

    for b2=0 to 17
        Lookup b2,["R","e","p","a","i","r"," ","I","n"," ","P","r","o","g","r","e","s","s"],b3
        x=b3
        gosub send_data
    next

return

'*************************************************************************
```

FIG._15E

```
Complete_Mesg:

      Gosub CLR
            low RS
            x=LINE1+5                              'center tect on top line
            gosub Send_data
            high RS

for b2=0 to 14
Lookup b2,["R","e","p","a","i","r"," ","C","o","m","p","l","e","t","e"],b3
x=b3
gosub Send_Data
next
return
.....
      Clr:
            low RS
            outl =%00000001          'Clear the display.
        pulsout E,1
            high RS
      return
.....

      send_data:
            outl =x
            pulsout E,1
.....

GETSW_LOC:

if SW2 then SW2_ON_UPD
if NOT SW2 then SW2_OFF_UPD

      SW2_ON_UPD:
            SW2_LOC="B"
      return

      SW2_OFF_UPD:
            SW2_LOC="A"
      return
.....
```

FIG._15F

Flowchart Description Figure J90

Figure J90 is an operational diagram of the circuit board shown in Figure Design1_2.ckt]. In Item J90.10 the input selector switch is checked in order to determine which of the three possible operations (A, B, or C) is requested by the user. As denoted in J90.20 if operation A is selected the device branches to operation A. If operation B or operation C is selected, the device branches to either of the respective operations. If none of the operations are selected the device continues to wait for a selection.

Operation A

Upon startup, if the operation input switch is selecting operation A, the device switches the user hard drive BUS Address ID to a number other than 0. At the same time the master hard drive's power is turned on and the drive spins up recognized at ID 0 in place of the user hard drive. This function, as shown in flowchart item J90.40, can use any type of switching device, whether analog or solid state, to switch both the user drive ID and the master drive power. In [Design1_2.ckt] the devices used are electromechanical relays, which are switched with a Darlington transistor pair at the coil input. As shown in [Design1_2.ckt] relays RLY1, and RLY2 switch the 12 volt and 5 volt power connections to the master hard drive. This configuration uses 2 relays but the same task may be carried out using any number or configuration of relays, transistors, or other type of switching device. In another implementation, the hard drive power could have the ground connections all switched in order to turn the hard drive on and off. This implementation is shown in figure 22C [Design1.ckt]. When the master drive power and user drive address are switched, an optional message on an LCD screen (optional, but used in this implementation), appears along with an indicator lamp, which denotes the operation currently executing.

Operation B

Upon startup, when operation B is selected by the input switch, the device waits for X seconds before operation execution. This ensures that the user has time to abort and run under the normal operating conditions. Operation B as shown in flowchart item J90.30 waits for the specified time delay J90.60 and then Resets CMOS J90.110. At the same time an optional indicator lamp (LED B) is illuminated as shown in J90.100 and a message is displayed on the optional user interface (LCD) screen.

Operation C

Operation C is a normal operation mode J90.50 that allows the computing device to startup in a non-repair mode. Besides normal operation a message is displayed on the optional user interface J90.80 (LCD screen), and a normal mode indicator lamp is illuminated J90.90 (LED C)

After the selected operation executes the device loops, in order to check for a change in the operation selector switch. This is shown in J90.140, which performs the same function as J90.10. At this point, if the operation selector switch has not been

FIG._16

1. The invention described herein is hereafter referred to as the: "Backup and/or Repair System" – "Multi-User System" – "Virus-Proof/Hacker-Proof System" – "Hardware-Repair System" – "Freeze-Buster System" – "Net-Lock System" – "Anti-Theft System" "Entertainment/Communication System."

2. BACKGROUND AND HISTORY OF INVENTION

2.1 For the past several years we have been developing a new type of computer repair process to enable typical, non-technical, computer users to repair complex computer problems without effort, repair knowledge or skills. Our goal was to provide a way for the computer user to simply flick a switch, push a button, or speak a command, and have the computer take care of fixing itself, quickly, and easily, no matter what was wrong with the computer. We also wanted to be able to protect the computer from damage.

We constructed a prototype switching assembly and installed it in a computer, with the control switch located on the front of the computer to give the user easy access to the switch, and enabled a person to:

• Conveniently and simply push a button, flick a switch, or speak a command and perform our automated repair process. This switching system and related features that we initially developed for the repair process also enabled us to create several derivative inventions.

This switch triggered relays that were used to switch between data storage devices by switching power and/or device identities -- functions that were used by the repair process. We wrote repair scripts and programs to automate the repairs, so that the user didn't need to know a thing about fixing computers. The repair scripts made backups of the user's data storage device, reformatted the data storage device as necessary, and then replaced some or all of the operating system data, and/or other settings and/or data. These repair scripts and programs were loaded onto the prototype, and after a few adjustments, we got the contraption working.

For the first time, in conjunction with a combination of our repair scripts and programs, a user could just turn a switch on their computer and the computer would fix itself. Incidentally, we realized that the same invention we had made for the repair process also provided the ability to easily switch between one or more data storage devices, and also to switch groups of data storage devices. By using the ability to switch between data storage devices, we were able to construct prototypes that had one or more data storage devices dedicated to particular users, and that by turning a switch the computer would be set up for a completely different user. Additionally, this gave us the ability to construct computers that could be used by network administrators to rapidly switch between large numbers of data storage devices, and therefore, more easily create and update templates used in network repairs, perform software installations and "updates" of client computers.

The devices, methods, and technologies we developed for the repair process led to many new discoveries. We realized that many unique inventions could be derived from this technology. For example:

• A device that could automatically backup and/or repair data storage devices and software.

• A way to enable multiple users to share a computer, each having total privacy of data.

FIG._17A

• A way to enable a computer to have totally separate and independent and multiple operating systems and unique setups on one computer.

• A way to enable switching between multiple data storage devices.

• A way to enable a computer to switch to an emergency "startup" and/or "operating" device.

• A method of switching data storage devices so as to "quarantine" viruses, block hackers, and protect data.

The repair process evolved. We wanted to be able to enable a user to test and repair more data storage devices and components, and be able to switch to other components in the event of component failure. We decided to develop a computer in which any part and any software could fail and/or become corrupt, and that our invention could fix anything, (hardware or software) that went wrong.

We realized there was a broader spectrum of devices that we could switch, which led to yet other unique inventions that were derived from this technology, such as:

• A way for users to switch from components that have failed to components that function properly.

We also recognized that certain types of devices are prone to causing computer "freezes " that can be resolved by breaking and re-establishing one or more of the connections to that device. Therefore we developed:

• A method of switching the connection to devices off and then back on, as a means to reset the connection and break out of a "freeze."

As we considered the repair process we also wanted to prevent the computer from needing repairs in the first place. One source of computer damage is from viruses and hacking from the Internet. We realized that our switching process could also be used to solve concerns about network privacy and security, and a desire to enable parents to control children's access to the Internet. We decided to use our invention to switch "on" and "off" network and/or communications connections. During this development process we developed:

• A method of switching and/or locking a network/communications connection in the "on" or "off" position.

So we added more switching features, such as giving a user the ability to turn a switch on their computer and switch between entire sets of data storage devices, the ability not only to switch between data storage devices, but also to switch power supply, jumper cable connections, network connections, and any type of circuit or connection, and enabled the ability to reset hardware and software settings. We added the ability to reset PRAM; BIOS; CMOS; CUDA, and any other chip, board, and/or device that stores such information. We then decided to add the ability to switch between a data storage device and a circuit board, and the ability to switch between circuit boards. Then we added the ability to switch remotely, and for the device not only to be on a computer, but on any type of computing hardware. We added the ability to switch between logic boards and entire computing hardware systems, neural networks, etc.

FIG._17B

We kept adding more switching functions, until virtually anything on the computing hardware device could be switched locally or from a remote source, and interact with other devices that were local or at remote locations.

Initially we had just given the computer user the ability to perform the switching process by mounting the switch on the computer where it was easy for the user to reach the switch. But as time went on, we added the optional ability for it to be controlled: by a system administrator, other users, by network devices and appliances, automatically, locally, or remotely.

We created optional scripts and programs to enable computers and computing hardware to do new things with the new abilities that resulted from the new switching features. These scripts and programs enabled new methods of: repairing computers, switching between data storage devices and groups of data storage devices, switching back up systems, reformatting data storage devices, providing for emergency switching of data storage devices and circuit boards and computing systems, recovering data from back-up storage devices, etc.

For example, a user could sit at a computer and flick a switch located on or in the computer to fix a corrupt data storage device (or any device), and be able to flick a switch on or in the computer to change from one device to a different device, such as a logic board and/or ROM chip, and/or network connection, etc.

Here is a different example using our switching system, in conjunction with programs and/scripts, to perform a backup and repair process for a malfunctioning hard drive:

- Two hard drives can be connected to a "regular" computer.
- A single toggle switch can be mounted on the front of the computer, or any location.
- Both hard drives are controlled by our special switching system that can control their "Device IDs" and/or "master" and "slave" settings, and power.
- Hard drive 1 is a "typical computer user's" hard drive with an OS, a few applications, documents, and e-mail. Its device identity is set as the "master" hard drive.
- Hard drive 2 can be partitioned into 3 partitions: a) a "start-up" partition b) a "master" partition and c) a "backup" partition. Partition "a" is configured to be the "booting" partition.
- During "normal" use, our switching system switches hard drive 2's device identity to be a "master" hard drive, but our switching system also turns off hard drive 2's power.
- In partition "a" we have a perfectly functioning OS and related software to control the repair process. In partition "b" we have exactly the same OS and exactly the same applications as on hard drive 1, in a "pristine state" with no "defects". Partition "c" is blank at this point in the process because the machine is new and "backups" have not yet been made.
- A program that copies data from hard drive 1 to hard drive 2 can then execute periodic "backups" of the user's data (e.g. the "My Documents" folder, email) to partition c of hard drive 2. These backups may be uncompressed, compressed, or represented by an algorithm. The program can be stored on any device that can store data (hard drive 1 or a flash ROM chip are two examples).
- This copying procedure can be made possible, for example, by utilizing a program that initiates the switching process (which switches the power "on" to hard drive 2 and gives it the "slave" device ID) at periodic intervals (that could be adjusted by the user) in conjunction with scripts that copy the user's documents and email to the "backup" partition "c". Each backup that is created may be given a unique time/date stamp so that when the user needs his/her data back s/he can choose from multiple "backups."

FIG._17C

Various parts of the OS, application software, or any other data on hard drive 1 could then be damaged, deleted, corrupted, or destroyed. The hard drive could even have "bad blocks" and/or sectors, and/or even physical damage on the surface of the hard drive. The corruption of hard drive 1 could be so terrible that the computer could not even "boot up." We could then repair it:

- "Flip" the "toggle switch" and the following switching process occurs:
- Our switching system then switches the "device ID" settings so that hard drive 1 becomes the "slave" drive, power is then connected to hard drive 2, and hard drive 2 is switched to become the new device ID of "master."
- The script that creates "backups" could then execute again and copy all or selective data from hard drive 1 (for this example, it could copy the user's "My Documents" folder and/or email.) to Partition "c" of hard drive 2.
- Another script could then run on hard drive 2 that would completely reformat hard drive 1. It could also map the "bad blocks" on the hard drive.
- Then another script could run on hard drive 2 that copies the "perfectly functioning" copy of the OS and applications from partition "b" over to hard drive 1.
- Optionally, a script could then run that asks the user which copy of the "backups" from partition "c" the user would like to revert to. Upon choosing a particular backup, the "My Documents" folder and email (from partition "c" on hard drive 2) would then be copied back to hard drive 1.
- When it finishes copying, another new script could run that shuts-down the computer and its hard drives.
- The user could then "flip" our switch again and it would switch the "device ID" back to "master" for hard drive 1, switch hard drive 2 to "slave" and cut its power.
- The user could then restart the computer, and it would boot-up perfectly into the freshly reformatted hard drive 1, with its freshly copied OS and applications, its freshly copied "My Documents" folder, and its malfunctioning behavior would thus be repaired.

We added optional switch locks to the hardware, and scripts and programs were developed to enable multiple users to use a computer with completely different sets of data storage devices, with each user unable to access the others storage devices. The functions of the scripts and programs can also be handled by programs in ROM, programmable logic controllers, circuit boards, and various forms of data storage devices. Numerous scripts and programs were developed, and we continued to develop the hardware.

In some situations, we soon learned it could be dangerous or cause damage, to turn a switch when power is being supplied to a device, and thus we invented several methods of preventing (or locking) a switch from turning or deactivating the switch when power was being supplied, and releasing the switch (or allowing the connection to be made) when it was safe and power was removed. This dangerous situation can also be avoided by using a logic controller to prevent a switching process from taking place until it is "safe." This would be accomplished by having the controller keep track of power, device identities, etc. and thus control switching as needed.

It then occurred to us that we could trigger various functions very similar to what we had developed via hardware by using software that could be integrated into: data storage devices, ROM, programmable logic controllers, circuit boards, etc., which led to a whole new approach to the repair process. To clarify this, in addition to software triggering the switching process, we found a way to replace most or all the switching process with software, or to use the software switching replacement in conjunction with the hardware switching.

FIG._17D

We then developed numerous options for triggering the process, instead of just being limited to a switch we realized it could be controlled via voice commands, via telecommunications, optical control, wireless communications, scheduled events, logic control, etc. We decided to provide optional security over the switching process not only by key switches, but by voice id, retina id, optical recognition id, thumb prints, voice prints, magnetic cards, passwords, encryption, and any type of method available to provide secure identification.

Therefore, we realized that we needed to make our invention work for a wide spectrum of users, no matter what options they wanted. We decided to make our invention scaleable -- expandable as needed -- with modular hardware components that could be quickly connected together if desired, and optional software scripts that could be mixed and matched as needed. Thus, if someone wanted one component switched, our switching system would consist of the switching circuitry needed for that one component and the optional software for that one function could be provided. If a user wanted more components switched, we would have the ability to just snap in additional modular components as needed, and provide the optional software for the additional functions. (Snapping the components together is optional... they can also be mounted as separate non-modular components.). Non-modular versions can also be produced as needed.

We wanted to make these innovations usable for robots, robotic devices, electronic devices, network appliances, vehicles, and all situations where computing equipment was integrated with other equipment and devices. Therefore, we revised the trigger and switching system so that instead of being limited by the restriction of having the switch and/or "trigger" located on the front of a computer (or under an access panel/door of some sort) the trigger and/or switch could be movable so that it could also be located anywhere to give the user(s)/operator(s) easy access. For example, in situations where a computer is used as part of a vehicle, robotic device, etc, the switching system(s) and/or trigger(s) can be located within easy reach of the user(s). It can be built into a dashboard, control panel, robotic control system, control room, house, building, door, remote location, or anywhere else.

We have ended up with a scaleable switching system with scaleable hardware and software and scripts and programs that perform many scaleable functions, that can fit together in modular fashion as needed; as well as, several methods of controlling the switching process.

We also invented electronic connections on the outside and inside of the computer that provide ways to connect lighting, decorations, LCD's, LED's, sound, and or video, to communicate what is taking place, and for entertainment purposes.

We also invented a light emitting/reflecting acrylic device that may also utilize/interact with the other inventions.

Upon evaluating what we had developed, we observed that it can be used to enable one or more functions described herein. Therefore our work evolved into writing this preliminary patent application.

We have attempted to put these inventions/processes/systems into 7 groups. They all utilize a variation of our "switching system" in conjunction with software and hardware. They are:

"Backup and/or Repair System"
"Multi-User System"

FIG._17E

"Virus-Proof/Hacker-Proof System"
"Hardware-Repair System"
"Freeze-Buster System"
"Net-Lock System"
"Entertainment/Communication System."

3. How are these inventions different from existing products?

3.1 The "Backup and/or Repair System" - new and unique features:

- Push a button (or other trigger to initiate the switching process) and the computer repairs a corrupt or malfunctioning data and data storage device(s) by executing a process of automatic steps (without technical skills needed by the user.)

- A combination of hardware and software processes that back-up data and can repair data storage devices (for example, automatically re-format a data storage device, and then restores a backup).

- Use of "master copies" (or "templates") of software programs on separate data storage devices for purposes of repair.

- The ability to "start-up" from a separate data storage device as needed to be able to function and/or initiate repairs.

- The ability to back-up data to a separate, internal, protected (because it may be switched or "un-mounted" or locked) data storage device without user involvement in order to enable repairs (such as reformatting) to the malfunctioning data storage device.

- The ability to backup, copy and restore data (and execute scripts and/or programs) so as to repair data storage devices, even if the data storage device is corrupt and "unbootable" by booting from a different data storage device.

- The ability to repair devices that utilize computer equipment without needing to access the computer equipment. Several other new features have been developed and are summarized in the abstracts that follow.

- Ability to utilize and/or be integrated with our other systems.

- Ability to automatically repair any type or model or brand of computing device.

3.2 "Multi-User System" - new and unique features:

- Push a button (or other trigger to initiate the switching process) and the computer switches "device identity" settings for data storage devices so as to enable different devices to become "startup" devices (for example, to accommodate manufacturer's default requirements for boot sequences).

- Ability to utilize and/or be integrated with our other systems.

FIG._17F

Ability to change device ID settings and/or power connections on data storage devices quickly and easily.

3.3 "Virus-Proof/Hacker-Proof" System - new and unique features:

- A means to automatically switch data storage devices by executing a process of steps (without technical skills needed by the user.) By switching device IDs and/or other means, so as to "quarantine" data, for a period of time, into separate data storage device(s). This allows time for updated virus programs to check for new viruses and "kill" them", and/or so that infected files can be accessed without contaminating the rest of the system.

- Ability to utilize and/or be integrated with our other systems.

- A means to prevent access to data within a computer by shutting off/on network connections as needed (automated) and by only having a data storage device "open" to the network that contain no "user" data or software.

- Automated web-site protection: data storage devices that "host" the data of a web site could be duplicated on many data storage devices and the "cycled" to be connected or disconnect to the network. In this way, "hacked" data could be repaired when it was cycled to the "offline" position, while an "un-hacked" duplicate data storage device could be instantly rotated "online" (switched) to take its place.

3.4 "Hardware Repair System" - new and unique features:

- Push a button (or other trigger to initiate the switching process) and the computer switches from malfunctioning or "failed" components by executing a process of steps (without technical skills needed by the user) to components that function properly..

- Ability to utilize and/or be integrated with our other systems.

- Push a button and the computer switches to a "backup" computer (internal components are all switched at once to duplicate components).

3.5 "Freeze-Buster System" - new and unique features:

- Push a button (or other trigger to initiate the switching process) and the computer momentarily interrupts the connection of a computer peripheral by executing a process of steps (without technical skills needed by the user) so that computer can "recognize" the device.

- Ability to utilize and/or be integrated with our other systems.

3.6 "Net-Lock System" - new and unique features:

- Push a button (or other trigger to initiate the switching process, for example, turn and lock with a key) and the system interrupts the connection of a network/communication connection (for example, the Internet and/or intranet) by executing a process of steps (without technical skills needed by the user) so as to restrict or allow access to said network/communication connection.

FIG._17G

- Computers (and other computing devices and/or hardware) don't have built-in switches that can be easily controlled by the user(s) (or by an administrator, or across a network, or wireless, or other method of control) that allow switching on/off of network and/or communication connections. The "Net-Lock System" provides these abilities and comes with optional scripts and programs that enable users to more easily use these new abilities. It also enables high speed automatic switching of network communications connections.

- Ability to utilize and/or be integrated with our other systems.

4. ABSTRACT

4.1 Overall Abstract:

The following did not exist until now: the ability for a user to control the switching process by locking/unlocking a switch on their computer, remote switching as we have developed it, software switching as we have developed it, the many items that our switching process can control, and our scripts and programs executing in particular orders with specific and unique, and useful results. We combined them into a switching process that provides many new functions.

The unique "Switching System" is an integral part of what enables these new, unique and useful functions to be made possible. The optional scripts and/or programs have been utilized in specific sequences to perform specific tasks in conjunction with the Switching System. The Switching System is more specifically defined in section 7.

The Switching System includes a switching process that can switch one or more of the following: between data storage devices, data storage device ID's (often controlled by jumper cables and/or a "cable select" cable on a data storage device), between power supplies, between jumper connections, etc. It can also reset PRAM; BIOS; CMOS; CUDA, and any other chip or board and/or device that stores such information. Any circuit between any two devices that can be "opened" or "closed" in any combination can be "switched". It can switch indicators (such as LED lights) for activity, power, and identity. It can also, reset hardware and software settings. As needed, it can utilize optional scripts and programs to enable use of these new switching abilities. It can do any of these switching processes in any combination and permutation.

(Please note that that the switching System can be scaled so that it can be used for one or more of the functions described in the following abstracts. It need not be constructed for more than one function, and can be scaled to perform any combination of functions as desired.)

4.1.1 Abstract of Software as a Switch "Replacement:"

Optionally, the "Switching System" can use a software "Switch Trigger" (see definition) replacement. In other words, a program and/or script can initiate the switching process, instead of a person pushing a button (see examples in section 7.1) The software can also replace a mechanical switch.

4.1.2 Abstract of Switch Lock, Bypass, Delay, and/or Cylinder Lock Device

4.1.3 Optional Switch locking mechanisms and switch deactivation mechanisms were developed to prevent damage to computer equipment if someone turned the physical "Switch Mechanism"

FIG._17H

at the wrong time. The locking mechanisms prevent the "Switch Mechanism" (a physical switch, for example, a "switch-lock") from physically turning when it is not safe to turn the switch. The switch deactivation mechanisms deactivate all or part of the Switching Process until it is safe to perform the switching process, e.g., when no electrical current is flowing to the devices being switched.

4.1.4 A logic controller can also be used to control the Switching Process and/or power, and/or any other connections and/or settings, so as to prevent damage to computer equipment by deactivating the Switching Process when it is not safe to switch.

4.1.5 Abstract of system Scalability and Expandability:

The Switching System switch can be scaleable and expandable. It can switch one item, device, or circuit, or very large numbers (theoretically infinite numbers) of items, devices, circuits, and/or computing hardware systems. It can be modular, and/or non-modular.

4.1.6 Abstract of Location and Accessibility:

One of the features that makes the each of these systems different, is that there can be a "Switch Mechanism" (see definition) (a physical trigger) readily available to the user, and that the Switch Mechanism can be located anywhere, including on or near the computing device, and/or convenient to the user. For the first time, the devices being switched can be switched easily and conveniently. The.Switch Mechanism can be located in easy to reach locations such as the front or side of computing devices, near user controls of robotic devices, on vehicle dashboards, in control rooms, locally, remotely, and/or wherever it is convenient for the user/operator. This enables the user/operator to easily reach a (the) switch(es). The Switch Mechanism can take the physical form of any sort of a trigger; for example, toggle switches, buttons, switches, switch-locks, voice control, retina identification, card readers, magnetic key systems, and any sort of local and/or remote system used for triggering a switching process.

Please note that when switching takes place, it is not necessarily switching to/from devices on the local computing hardware, but may switch to or from devices on a network, and/or global computer network and/or communications network. The "Switch Trigger" may be located separately from the "Switch Mechanism."

*****

4.2 The "Backup and/or Repair" System Abstract

The "Backup and/or Repair" System consists of copying data, executing scripts, and switching.. data storage devices for the purpose of repairing data and/or data storage devices.

The "Backup and/or Repair" system consists of a series of software and/or hardware events, that, when strung together in specific order(s), back-up and perform repair functions of data storage devices.

This invention can be built as a computer that contains the features described herein, and/or can be built as independent devices that can be added to or integrated into computers.

FIG._17I

4.2.1 The "Backup and/or Repair" system as Used for data storage device repair:

One option is for the "Backup and/or Repair" system to keep and/or utilize one or more perfect master template(s) of the users' data storage devices, and it can back up and archive the user's data using software or scripts [(which can, for example, be located on a StorExecute (see definition)] that "backs-up" or copies data from one data storage device to another. When the data storage device has a "problem," the "Backup and/or Repair" system can use its switching features to access the perfect master template(s), and/or the backup(s) and/or archive(s), and may use some scripts and/or programs [(which can, for example, be located on a StorExecute (see definition)] to restore a computer to a functioning state. Rather than using a master template, the "Backup and/or Repair" system can also conduct the repair from a backup, and/or archive on a separate data storage device.

The following events can take place in variable sequences, and each step is optional, depending on desired results:

- The "Backup and/or Repair" system copies data from a user's data storage device to one or many different data storage devices.
- These other data storage devices would then contain "backups" (or duplicates) of some or all of the user's data.
- After the data is copied, a switching process can then protect the duplicates on the other data storage devices by restricting access to them (for example, by locking and unlocking a series of partitions, or by creating "read-only" files and/or folders and/or partitions.
- Then, if the user's "main" data storage device becomes corrupted, or it fails to "boot-up," a switching process can allow a different startup device to "boot-up."
- The user's data could then be copied BACK to the original user's "main" data storage device; and/or, a different data storage device can copy only the Operating System software, and/or application software to the original user's "main" data storage device.

An optional hardware diagnostic program may execute in conjunction with the switching system to administer repairs on demand and/or self repairs by switching from defective data storage devices to data storage devices that function properly.

Note: The user's own monitor could be used by the Repair System to communicate with the user. For example, a second video card (or other device to process a video signal) may be utilized by the system so that it can display information to the user on the users own monitor.

Alternatively, for example, the system may utilize the existing video card by switching its signal so that the repair system controller can control it and send its own video signal to the monitor (to display repair messages or information to the user using his/her monitor).

4.2.2 The "Backup and/or Repair" system as used for "Emergency Startup" & "Emergency Computing"

The "Backup and/or Repair" system can optionally be used for switching to a separate data storage device for use as an emergency startup system (note: the data storage device may be "on" and always available for instant switching, so that the computer does not require a restart). Thus, when a startup device is not recognized by the computing hardware, and/or at the request of the user, and/or other request from elsewhere, (or automatically when a problem is detected, and/or

FIG._17J

on a schedule) the "Backup and/or Repair" system can then switch to a secondary startup data storage device and restart, and/or switch between RAID mirror data storage devices, etc.

4.2.3 Abstract of the "Backup and/or Repair" system as Used for Formatting and Device Testing

The "Backup and/or Repair" system can optionally be used to test data storage devices. It can be used to temporarily switch to a data storage device for startup, while a data storage device reformats and/or tests another different data storage device.

4.3 "Multi-User" System Abstract

"Multi-User" System as used to enable multiple users to share a computer, each having total privacy of data:
The "Multi-User" System can be used to enable multiple users to use computing hardware as if each of the users had their own private computer. When a user is using a computer, the "Multi-User System" sets up the data storage device, operating system, applications, etc. just for that particular user. Then, when a different user wants to use the computer, the "Multi-User System" hides away the previous users data storage device, operating system, software, and data, and provides a different data storage device, operating system, and software for the new user.

4.3.1 Abstract of the "Multi-User" system as used to enable a computer to have totally separate and independent and multiple operating systems and unique setups on one computer
The "Multi-User System" can be used to control totally different setups of operating systems and software, and switch back and forth between them. For example this would enable a computer to be set up with Linux and movie editing software in Japanese, and the "Multi-User System" could then switch the computing hardware to be set up with Windows and mathematics software in German.

4.4 "Virus-Proof/Hacker-Proof System" Abstract

"Virus-Proof/Hacker-Proof System" as used for a method of switching data storage devices so as to "quarantine" viruses, block hackers, and protect data.

By switching data storage device IDs and/or power, and/or network connection, and/or other means, data storage devices can be connected to a computer in such a way that one or more of the data storage devices can be isolated from other data storage devices, and/or isolated from network connections.

Thus, if a hacker or virus were to enter a data storage device that was connected to a network, said hacker or virus could only access one of the data storage device(s) because the other data storage devices were "separated" by the Switching System. This data storage device could be devoid of user data. Incoming data, such as email and/or web downloads could then be transferred to a data storage device that acts as a "quarantine are". After a period of time has passed, and/or virus checks have been run, data can then be safely transferred out of "quarantine" and onto user's "regular" data storage device. In this way, a data storage device could act as a "quarantine" for data. Viruses could then be "killed" in this isolated data storage device at a later time (for example, it would give a user enough time for an anti-virus definition to be written and downloaded for a new virus. This system would also prevent infected files from being accessed without contaminating the rest of the system.

FIG._17K

Also, a user could switch from one data storage device that was "private" or "isolated" (by using our switching system) from the network, to a different data storage that was exposed to the network, but was "empty" of user's personal data. Thus, a hacker would not be able to access the user's private data storage device., but only the one containing no user data.

4.5 "Hardware-Repair" System Abstract

"Hardware-Repair" System as Used for Computer Repair:
By utilizing the Switching System, a user can easily switch from components that have failed to components (and or computing device) that function(s) properly. An optional hardware diagnostic program may execute to determine the status of hardware, and if necessary, switch from defective devices to devices that function properly, (and/or to a secondary computing system) by utilizing the Switching System. Note: The switching of data connections may also occur while electrical power is still connected to the device(s) (see section 7).

4.5.1 Abstract of the "Hardware-Repair System" as Used for Formatting and Device Testing:

The "Hardware-Repair System" can optionally be used to test data storage devices, circuit boards and computers. It can be used to temporarily switch to a data storage device for startup, while a data storage device reformats and/or tests another different data storage device. It can be used to switch to a different logic board, network connection, or computing system. It can test the hardware that was taken out of use. One way it can do this is by switching from the device in-use, to alternate devices that are used while testing and formatting takes place.

4.6 "Freeze-Buster" System Abstract

"Freeze-Buster" System: a method utilizing the Switching System to momentarily switch the connection(s) to devices "off" (not connected) and then back "on" (connected), (or vice versa) as a means to reset the connection and "break out of" a device and/or computer "freeze."

4.7 "Net-Lock" System Abstract

"Net-Lock" System: a method of "locking" a network/communications connection to be either "on" or "off". This method utilizes the Switching System to be able to switch and/or lock a network and communications connection both "on" (connected) or "off" (not connected). The "Net-Lock System" may also contain optional scripts/programs that switch and lock the network and/or communications connection (including wireless).

This led us to another evolution. We realized that companies would want a way of shutting off internet connections, without shutting off their own intranet. So we developed methods for switching from an Intranet connection to a full web connection, and visa versa. This may also be done by utilizing two network connection devices that can be independently switched. This can also be achieved by integrating two network "jacks" (and associated and necessary circuitry/parts) onto one network interface card that can be independently switched.

Thus, if someone wanted one network connection switched, our switching system would consist of the switching circuitry needed for that one connection and the optional software for that one function could be provided. If someone wanted more connections switched, we would have the ability to just snap in additional modular components as needed, and provide the optional software

FIG._17L

for the additional functions. (snapping the components together is optional... they can also be mounted as separate non-modular components.). Non-modular versions can also be produced as needed.

The "Net-Lock System" can also switch on and off network connections. It can turn off and on (and lock and prevent access to) connections to a global computer/communications network, Intranet connections, and all other types of network connections.

For example, a parent could connect over a global communications network to their home computer and turn off and lock their computer network connection from use... and only they can unlock it.

Optionally, the user can restrictively control the "Net-Lock System" settings. For example, software could provide options such as: who has the right to "lock" or "unlock", when unlocking can occur, (on a schedule, and/or certain hours, etc,) etc.

An optional program/script gives the user/operator locking/unlocking the network and or communications connection. Optionally, this process could also run automatically.

5. Explanations of Terms

5.1 Master Template & Master Template Storage Device & Master Template System

5.1.1 Master Template: A master template is a collection of software that consists of one or more of the following: operating system, applications, or whatever else the user and/or maker of the template wants to put on it, and/or default/user preferences and settings. It is created by copying said software onto a data storage device (or partition) that is defined as a Master Template Storage Device.

5.1.2 Master Template Storage Device: A data storage device and/or partition(s) that is/are used for storing Master Templates. It may exist locally, and/or over a network. Optionally it may consist of more than one data storage device. This may be permanently or temporarily set to "read only" access during the repair process. Optionally, it may also be set to "read-write" access, for example, during an update.

5.1.3 Master Template System: A master template is created, copied, or exists and is stored on a Master Template Storage Device. It may be updated on demand, and/or at specific intervals defined and executed by a program. Data may be installed (and/or optionally copied), thus creating an updated master template.

5.1.4 StorExecute: Any method and/or device for storing and/or executing a program. This can be a logic board, CUDA, EPROM, chip, circuit board, etc.
5.2 Backup & Backup Storage Device & Backup System:

5.2.1 Backup: A copy of data to another data storage device and/or partition. It exists on a data storage device that is defined as a Backup Storage Device. This may be permanently or temporarily set to "read only" access immediately following the backup process. Optionally, it may also be set to "read-write" access, for example, during an update.

FIG._17M

5.2.2 Backup Storage Device: A data storage device and/or partition(s) that is/are used for storing backups. It may exist locally, and/or over a network. Optionally it may consist of more than one data storage device.

5.2.3 Backup System: Optionally, "on demand," and/or at specific intervals defined and executed by a program, a backup is created (as needed). Backups may be overwritten on demand and/or based on a computer program that, for example, but not limited to this example, staggers the intervals for creating backups over time, thus creating a collection of backups. Optionally, backups may be created by currently existing programs. Optionally, backups may be incremental, differential, or full. The backup Storage Device may also have multiple partitions that are turned into "read only" partitions immediately after that user's data has been copied there, and could cycle back over the oldest partitions first, after all partitions were filled.

Optionally, the Switching System may switch off the accessibility of the Backup Storage Device (by interrupting power and/or changing device ID settings, or other means) so that said data storage device is only accessible while backups are being created or restored, and thus hidden from the user the rest of the time.

Several different examples follow:

A program or script runs a "copy" command that makes a full, or partial, or compressed, and/or "normal", and/or encrypted, and/or "read-only" backup of User's data. This can be done by using any one of a number of software backup, or cloning products such as, but not limited to Retrospect; Assimilator; Backup Exec, Ghost, File Wave, etc., or by using a "copy" command, or running a block by block copy of the User's data.

Another Example: Backups could be stored that are: same day most recent, same day next older, same day next older, 1 day old, 2 days old, three days old, one week old, two weeks old, three weeks old, one month old, two months old. As time goes by, obsolete (oldest) backups may be overwritten. They can be made locally, or over a network.

Repair Examples

Example 1: A second operating system runs in the background with hardware and/or software process watchers that watch for a freeze, and/or any type of problem. When a problem occurs the repair process replaces any files and/or data that is different/modified/etc. as needed., resets connections as needed, and may copy and/or reset RAM, information in the processor, etc. as needed, and send a delayed copy of the data to the processor, prior to the last event that may have caused the problem (such as opening of an incompatible application).

2) Example 2: A second and "mirrored" operating system can run on a second processor, and a third another processor (or microprocessor) can run a repair program. The second data storage device can be running in a time delayed mode. When a problem occurs, the process in the second processor is stopped, (before a freeze), the freeze is analyzed, and the processor and ram is set up as they were before the event occurred that may have triggered the freeze.

Alternatively, it could backup in time without necessarily (or in conjunction with) analyzing what went wrong, and see if the freeze occurs again... if so, "back up" further in time... until such time as the freeze does not occur.

FIG._17N

Comparative Copying and replacement of files and data.

Master template may be compared to the User data storage device and/or to a partition and/or area of the user data storage device that it needs to repair. Some or all of the data is repaired by copying from the master and replacing software on the user data storage device that is different, changed, modified, or missing. Any items that do not match the template are removed.

Shortcuts, finder flags, aliases, icon positions, folder views, etc. are set to match those needed on the user data storage device. Based on user preferences and/or defaults, particular items on the master template can be marked or selected as items not to delete and/or items not to copy.

The master template may also contain a log and/or database of information to store information, and during the repair process set such items as the volume name, machine name, user name, password(s), which master template to use, which Data Storage Device, and/or partition to use, etc. These may also be adjusted by the user as preferences.

There can be various (and different) master templates to choose from. When one or ones to be used can be selected by the user(s) in preferences. Example 1: a user named "Mary" may use one master template, whereas a user named "Fred" may use a different master template. Example 2: A robotic device should be controlled a particular way in a particular situation. Thus it is using "template A". The situation changes and it is decided that the robotic device should act a different way. A switch may be made to utilize master "template b". Any number and variety of master templates may be utilized. Example 3: In the anti-theft version, the master template can be shifted to be a "bogus" template if the ID check fails. Which user data storage device(s) to repair can also be a "default," and/or a preference, and/or can be modified by an administrator. More than one user data storage device can be repaired.

The master template(s) and data storage device(s) can also be repaired. In this case (a) different master(s) (perhaps on a network) can be used, and the repair process can be similar to the options described herein for the repair process of the user data storage device, except it is the master template(s) and data storage device(s) being repaired.

User preferences can enable a user to select whether only part of a data storage device is repaired, and/or if the entire data storage device is repaired (such as reformatted) but if only some data is replaced, for example, if just the data on "Partition A" is replaced, but not the data on "Partition B".

A sub-master (a subset of a master) can also be used as all or part of the repair process.

In preferences, data on the master can be selected for special attention such as: always replace, never replace, replace if changed, replace if missing, etc.

A database can be used to store preferences, and information about which template to use, which user data to repair, etc.
Example of one embodiment of the repair process: When repair is needed on the primary system and/or data storage device, switch to secondary operating system that may be mirroring and/or having delayed mirroring (and may use a secondary processor and/or protected processing), do repairs on primary system, reboot primary system if needed, switch back to primary system.

FIG._170

The repair process can function in a number of different ways. For example, it can be user controlled, it can always take place on a schedule, and/or on startup, shut down, etc.

The master template may be a "perfect" installation of the system and software and/or data that a user wants and/or is required and/or desired on their computer/computing device, that may also have been checked for conflicts and/or errors and said errors could then corrected by an IT professional.

There are a number of ways to create a master template. For example:
an original "perfect" installation (where errors may have been identified and corrected) can be made on the user data storage device and then copied, or installed, to a second data storage device that contains the master template.
It can be created on another computer elsewhere, and downloaded via a network to reside on the users' computers.
It can be created and/or reside on a data storage device located on a computer elsewhere, and run across a network to repair the users' computer.

It can be created on the storage device and/or partition used to store the master template.

When the master template is created elsewhere and "run", or installed, over a network, or created on a different data storage device than the user data storage device, then shortcuts and/or aliases may need to be modified to work properly when they are copied to the user data storage device. In this case, during the copy process, the code fixes those shortcuts and/or aliases to point them to the correct item on the user data storage device.

If a reboot of the computer is required, this step can be eliminated by copying the startup sequence of the computing device to an EPROM, and keeping it powered "on" when a computer is "shut down". RAM and/or other volatile memory, (and/or a time delay version of RAM and/or other volatile memory) can also be stored, and copied back as needed.

Optionally, one or more additional switches and/or switch triggers can be utilized by user to confirm the repair process.

5.3 Archive Storage Device:

5.3.1 Same as a backup (5.2), but archives are never overwritten. Often (but not always) the type of media used for making an archive can only be written once.

5.4 Startup Device

A startup device is any device that can be used to "boot" or startup up a computer and/or computer equipment. It may exist on any media, device, and/or circuitry that can perform this function, and can be performed across a network.

(Intentionally no section 6)

7. Switching System:

7.1 Switch Mechanism: a physical switching device or a software switch (which uses software

FIG._17P

programs) to execute instructions and/or events. If a hardware switch is used it may be mounted on and/or in the computing hardware and/or on devices, control panels, dashboards, remote locations, control rooms, and anywhere.

For example, consider two data storage devices: Device A and Device B. The "Switch Mechanism" (in the form of a physical mechanism) could initiate the change of the device ID of Device A, and change the device ID of Device B. If the devices use jumpers for ID switching, the switching process could switch the jumper(s) on devices A and B to different IDs.

Software can do the same thing. For example, a program stored in a StorExecute (see definition) can be used to tell the computing hardware to "un-mount" Device A and "mount" Device B and treat the identity of Device B as a different ID.

7.2 Switch Trigger: Any method of triggering a Switch Mechanism to initiate the Switching Process. For example: Turning a switch or key, voice command recognition system, optical recognition system, software command, and/or any of a myriad other ways of triggering a switch to occur. The switch trigger can be controlled by the user or operator locally and/or remotely, across a network, wireless, etc. The switch trigger may be mounted on and/or in the computing hardware and/or on devices, control panels, dashboards, remote locations, control rooms, and anywhere. It may also be initiated in software only.

7.3 Switching Process: The process of using a Switch Mechanism to switch data storage devices to and/or from other data storage devices.

The Switching Process may also use an optional program/script to give the user/operator the choice of which data storage device they want to use. For example, a dialog can be used to give the user the option to choose from an array of several data storage devices. Optionally, this process could also run automatically. This process may include the opening and/or closing of electrical circuits connected to data storage devices, and/or ground connections, and or jumper cable connections, to switch which device "boots" the computing device, and/or switch data storage device identity.

Switching Process: The ability to switch device(s) and/or circuitry to and/or from other device(s) and/or circuitry.

An optional program/script and/or switch(s) gives the user/operator the choice of which device(s) they want to switch. For example, a dialog can be used to give the user the option to switch from "parent data storage device" to "child data storage device"; or offer an array of several data storage devices to choose from; or the user or program can switch to a different logic board or RAM chip, circuitry or computing device. Optionally, this process could also run automatically.

7.3.1 Switching Process: The ability to switch device(s) and/or network and/or communications connections.

7.3.2 The switching of data connections may also occur while electrical power is still connected to the device(s).

7.4 Switching a Data Storage Device:

FIG._17Q

The switching system may perform one or more of the following functions:

7.4.1. The switching process can change the device ID for a data storage device. For example, in the case of a hardware switch mechanism, in some cases it is a matter of closing or opening the circuit(s) of the jumper(s) for one ID, and not closing/opening other jumper(s) for different IDs. Another example: switching a cable select cable so that the "master" becomes "slave" and "slave" becomes "master" is another way of switching device IDs.

7.4.2. The switching process can change the device ID for a data storage device in other ways: this could also be achieved by having one data storage device (device "A") set to a particular ID (that is the startup device ID), but with its power switched off. Another data storage device ("B") could be in normal use. Then, when "A's" power is switched "on", this device, along with its ID, becomes "active", thus making it ("A") the new start up device. The ID used by device "B" can be switched to a different ID.

7.4.2.1. The Switching Process can switch data storage device power on and off, thus "hiding" and "un-hiding" the device; switching between making the device inaccessible and accessible.

7.4.3. In the case of software, a program may decide which device to mount and/or which device(s) to treat as active, and which device(s) to treat as inactive.

7.4.4. A software version may execute a program in a StorExecute , which can mount one data storage device and "un-mount" a different data storage device. Additionally, device Ids could be switched in software

7.4.5. Another example of how a software switch could operate is for a StorExecute , to be programmed to give the user and/or operator the choice which data storage devices and/or startup devices to use, and/or be programmed to make scheduled changes between devices in use. A StorExecute , can also change to become the default data storage device(s), unless the data storage device doesn't mount or freeze repeatedly, in which case it can automatically switch to another startup device and/or set of data storage devices.

7.4.5.1 Functionality of Circuit Board

This is but one example of how the Switching System can be controlled.

In circuit board figure Fig._22T:

• A jumper shall control whether the time delay circuit receives power. Thus a jumper can disable the time delay circuit.

• If power is being supplied to power control indicators A, or B, or C, do not allow the board to switch anything, even if power to a power control indicator is changed. For example, if power is being supplied to power indicator A, and then is switched to power indicator B while power is still being supplied to indicator A, ignore the change and keep the terminals closed that are associated with power control indicator B. Only if power is removed from all power control indicators for a period of 3 contiguous seconds or more, allow the changes in which circuits are closed when power is restored to one of the power control terminals.

FIG._17R

• Boards and switches combined must be any of these sizes or smaller (smaller is better):

• The socket shall bypass the circuit board with the neutrals (see diagram of neutral jumper bypass).

• If Circuit A: Wait X contiguous seconds and check to see if power is still on to A. If so, turn on time delay circuit. Leave circuit on Y seconds and then turn off circuit. If power is still on to B, don't run again until such time as power is removed from all controls A and B and C for at least 3 contiguous seconds.

• If circuit C, turn on time delay after X seconds (follow b above) and then turn on 1,2,3,4.

• ID Jumpers 4 and 6 are optional spares.

• For multiple users/operating systems, and/or data storage devices, duplicate the circuitry in the drawing (except for the controller and switch/switch lock... in most cases only one controller and switch/switch lock is needed.

In reference to figure Fig._26K:

Please note that this is just one example of the "Backup and/or Repair System" and is a functional design. The integrated "Backup and/or Repair System" Switching System can be integrated anywhere in or on the storage device, and/or integrated into the storage device circuitry.

It can be built with wires and relays, and/or a software switching process, and/or a circuit-board, and/or a logic control device. An optional switch can be used that is separate or integrated into the data storage device.

7.4.6. An optional program/script can give the user/operator the choice of which backup and/or archive they want to use to the repair. Optionally, the backup and/or archive process could also run automatically and/or on a timed schedule.

(Intentionally no section 8)

9. How each System works:

9.1 The Switching System:

Four issues should be clarified when considering the Switching System. The way in which the "Switching Process" is initiated ("Switch Trigger"), the hardware (or software) switch ("Switch Mechanism"), that which is being switched (data storage device IDs, power, etc.), and the location and/or accessibility of the "Switch Mechanism".

A Switch Trigger is used to initiate a hardware or software switching process that enables a computing device to switch between one or more data storage devices and/or partitions on one or more data storage devices; and/or sets of data storage devices (Fig._24E). When using a "hardware" switch (i.e., when the "Switch Mechanism" is a physical device, as opposed to software only), the switch can switch power and/or device ID jumpers and switch between data storage devices and/or groups of data storage devices, as needed. Similar results can also be achieved

using software instead of, and/or in conjunction with, the hardware version of the "Switch Mechanism".

If a "hardware" switch is used, it may be mounted on and/or in the computer and/or computing hardware and can be controlled in any way switches are controlled, including but not limited to switches, switch locks; voice control; optical recognition, encryption, passwords, etc. The switching device can be controlled by the user or operator locally and/or remotely, across a network, wireless, and/or automatically, etc.

The Switching System takes place as described in section 7. Optional scripts/programs designed for the switching process function can be executed as well. See examples described in section 7.3.6.

9.1.1 Switch Lock, Bypass, Delay, and/or Cylinder Lock Device - How it Works:

The locking mechanisms and switch deactivation mechanisms were developed to prevent damage to computer equipment if someone turned the switch at the wrong time. The locking mechanisms prevent the switch (or switchlock) from turning when it is not safe to turn the switch. The switch deactivation mechanisms deactivate the switch until it is safe to use the switch. Some versions of the Switching System do not require a switch locking mechanism or switch deactivation mechanism.
There are several ways to construct the lock. It could be done with a brake, clutch, tooth, clutch, solenoid, piezo device, etc. Two examples follow:

9.1.2 An optional solenoid, brake, lock, or clutch assembly can be used to prevent a key from turning in the switch lock when power is on. (i.e. Fig._24A, Fig._24B, and Fig._24C). This device can stop either the lock cylinder from turning, and/or a rod and/or other attachment to the switch and/or switchlock.

Example: A mechanical rotary cam lock with an extension or mechanical key lock with an extension shaft is mounted so that the knob or key can be turned from the outside of a computer, or can be located behind an access panel that can be accessed easily by the computer user.

Coming from the rear of the cam lock or key lock is a round extension shaft, 1/4" in diameter, for example. The shaft is surround by a brake, such as a friction brake, magnetic brake, or many other types of brakes or clutches that can be used to stop a shaft from turning, the shaft then is connected to a electric switch. These three devices can be separately mounted on a platform, or to make things a bit more compact, they can be combined into one switching device with one housing.

When power is on, power is supplied to:

1) the brake so the switch can't be turned, and

2) power is provided to the input power post of the main electric switch.

Therefore wherever the switch is positioned before the computer is turned on and power is supplied, devices connected to that circuit will get power, and at the same time because the brake is on, the user will be unable to turn the switch to another position until the power is off.

FIG._17T

Example: Instead of locking the shaft, the shaft can be eliminated, and the brake/clutch/solenoid/piezo device, etc. can lock the cylinder and thereby prevent it from turning.

Or, another method can prevent damage to the computing device: Option: An "StorExecute" (logic control device, Programmable Logic Controller and/or circuit board) can be used that can perform one or more of the following functions: reset the PRAM, CMOS, BIOS, etc., switches data storage device identities as specified, and switches data storage devices as needed, switches their power as needed, restarts or resets the computer, deactivates switching process as needed, runs computing device as needed, etc. (Note: Switch Trigger can be initiated in myriad ways: the Switch Mechanism does not need to be physically moved to utilize integrated voice activated control, and control via telecommunications and network access – built into the circuit board/PLC. This board/PLC, etc., can also delay some switching events as needed, while other events transpire. For example if the user turns a switch the PLC/circuit board may delay that switching process, until it is safe for that event to take place without damage. Board may also deactivate Switch Trigger, and/or Switch Mechanism, and/or Switching Process as needed.

9.1.3 Switching System Scalability and Expandability- How It Works:

The Switching System switch is scaleable and expandable. It can switch one data storage device, or very large numbers (theoretically infinite numbers) of data storage devices. This is done simply by adding or removing switching circuits and control. We have developed modular switching devices than can be used for this purpose.

9.1.4 Location and Accessibility - How It Works:

One of the key features that make the Switching System different, is that there can be a "Switch Mechanism" (see definition) (a physical trigger) available to the user, and that the Switch Mechanism can be located anywhere, including on or near the computing device, and/or convenient to the user. For the first time, the devices being switched can be switched easily and conveniently. The Switch Mechanism can be located in easy to reach locations such as the front or side of computing devices, near user controls of robotic devices, on vehicle dashboards, in control rooms, locally, remotely, and/or wherever it is convenient for the user/operator. This enables the user/operator to easily reach a (the) switch(es). The Switch Mechanism can be easily accessed by any sort of a trigger, for example, toggle switches, buttons, switches, switch-locks, voice control, retina identification, encryption, card readers, magnetic key systems, and any sort of local and/or remote system used for triggering a switching process.

The Switching Process that can switch one or more of the following: between data storage devices, device ID's, power supply from one source to another, and/or on and off, and jumper connections. It can switch indicators (such as LED lights) for activity, power, and identity. It can also reset hardware and software settings. As needed, it can utilize optional scripts and programs to trigger the use of these new switching abilities. Another option, it can trigger a switching process over a network, and/or global computer network/global communications network (for examples see figures Fig._26A - Fig_26L, Fig._24E).

The Switch Trigger can be used to initiate the switching process that switches between one or more data storage devices and/or partitions on one or more data storage devices; and/or sets of data storage devices. When using a "hardware" switch trigger, the switch trigger can switch power and/or device ID jumpers to switch between data storage devices and/or groups of data

storage devices as needed. The Software Switch can accomplish similar results (see explanation under Software Switch).

9.1.5 Note: Any switching features listed below can be combined with other switching features.

9.1.6 Note: Scripts and programs can be located on a StorExecute. (e.g.: figure Fig._22D and Fig._26L)

Fig._26L: The series of instructions of these various systems can be integrated into a StorExecute. These methods may utilize software, hardware, and/or a combination of hardware and software. For example, a program can be written into ROM that controls the startup sequence and/or data storage device selected for startup during normal use; and then when a request for repair has been made, the program in ROM can switch the startup device. Additionally, for example, a memory module can contain the master template(s), the backup(s), archive(s), and/or any other software, instructions, and/or code to execute the Switching System.

In the figure Fig._26L, Fig._26M represents the concept that the switching system can be executed via a StorExecute. It may store and/or execute the some or all of the instructions that may control the entire repair process. For example it can reset PRAM, CMOS, BIOS, etc. settings as in L76.2, it can determine which data storage device will be the startup device, and switch startup devices during the repair process if needed, (it can also perform some types of repair without switching startup devices), it can control backups and archives to a storage device, memory module, a remote location, etc. It can also control communication with the users, manage automatic repairs and processes, including, but not limited to execution of all scripts, switching, programs, etc. It can interact with other logic control systems and devices to conduct, coordinate and manage collaborative repairs with other computing devices using the Switching System.

In the paragraph above, when we use the word "computing device" it is used in its broadest sense... for example, it could consist of biological and/or "nano" computing elements and switching mechanisms.

9.1.7 There are myriad combinations, permutations and variations on how hardware and software switching can be done. The following are only a few examples for demonstrating concepts. A person skilled in the arts can develop numerous variations in short order.

*****

9.2 Specific Examples:

9.2.1 The "Backup and/or Repair" System as Used for Computer Repair - How it Works:

The "Backup and/or Repair" system can be used to repair computers, and other devices that use data storage devices. One option is for the "Backup and/or Repair" system to keep and/or utilize one or more "perfect" master template(s) of all or some of the user's data stored on their data storage device(s), and it can backup and archive all and/or some of the user's data. When the data on a data storage device, or the data storage device itself has a problem, the "Backup and/or Repair" system uses its Switching Process and the Master Template(s), and/or the Backup(s), and may use a/various script(s) and/or program(s) which can for example be located on a StorExecute,

to repair the data storage device(s). Rather than using a Master Template, and/or in addition to using a master template, the "Backup and/or Repair" system can also conduct the repair from a backup, archive, and/or other data storage device(s). (Example: see figures Fig._21B – Fig._21O).

Optionally, the Master Template System and/or Backup System, and/or startup device may be utilized to perform their functions as described in sections 5.1 and 5.2. (Example: see figures Fig._21B – Fig._21O)

If a decision is made (by user and/or operator and/or program) to repair a data storage device (Example see figures: Fig._22B – Fig._22M), then the Switch Trigger triggers the Switching System. Several options and variations on what devices are switched can transpire at this point. This can occur in many variations. The Master Templates and Backups and Archives can exist on only one data storage device or on many different data storage devices in myriad combinations and repair one data storage device, or many.

Optionally, the Switching Process (See example figures Fig._20A – Fig._20J, Fig._24E and Fig._22B - Fig._22M) can change the startup device to a different startup device that is connected to the computing system hardware either locally or via a network. Upon starting ("booting") from this new startup device, and/or upon switching to another logic control device, a program that may reside on this new startup device, and/or a logic control device, and/or StorExecute, may execute and display a list of various options, for example, methods to "fix" the data storage device that is in need of repair, and/or update Master Template.

An optional program and/or script may also execute (or give the user a dialog box offering the user to select) functions to be performed such as: user may choose to reformat the data storage device that needs repair ... and/or the user may choose to execute a low level or quick reformat of the data storage device, or the user may choose to skip the reformatting process. When choice is selected, the function is performed by a program that executes that function, e.g., reformatting software is run.

Optionally, the user may also be asked if he/she wants to copy all or some of the data that exists on the Master Template data storage device(s), and/or Backup data storage device(s), and/or Archive data storage device(s), to the data storage device(s) that need(s) to be repaired. If this choice is selected, then the same type of program that was used to make the backup(s) could also be used to copy and/or modify the data from the Master Template back to the user's data storage device, thus making the data storage device used by the user (nearly) identical to the Master Template. (There may be some variations made to the data on the repaired data storage device as needed-- for example, shortcuts and/or aliases may need to be modified to point to their targets correctly).

Optionally, a program and/or script may also execute (or give the user a dialog box asking the user to select) any of the following functions to be performed: copy data from one data storage device (from a particular date or partition) to a different data storage device, as needed. For example, the same type of program that was used to copy User Documents, Email, Bookmarks/Favorites, Quicken data and/or preferences, etc. to the Backup Storage Device, can now reverse its process and copy all or some of the data back to the user's data storage device that needed repair. (A script may be executed that gives the user the choice of which backup (e.g. from which date) to copy the data).

FIG._17W

to repair the data storage device(s). Rather than using a Master Template, and/or in addition to using a master template, the "Backup and/or Repair" system can also conduct the repair from a backup, archive, and/or other data storage device(s). (Example: see figures Fig._21B – Fig._21O).

Optionally, the Master Template System and/or Backup System, and/or startup device may be utilized to perform their functions as described in sections 5.1 and 5.2. (Example: see figures Fig._21B – Fig._21O)

If a decision is made (by user and/or operator and/or program) to repair a data storage device (Example see figures: Fig._22B – Fig._22M), then the Switch Trigger triggers the Switching System. Several options and variations on what devices are switched can transpire at this point. This can occur in many variations. The Master Templates and Backups and Archives can exist on only one data storage device or on many different data storage devices in myriad combinations and repair one data storage device, or many.

Optionally, the Switching Process (See example figures Fig._20A – Fig._20J, Fig._24E and Fig._22B - Fig._22M) can change the startup device to a different startup device that is connected to the computing system hardware either locally or via a network. Upon starting ("booting") from this new startup device, and/or upon switching to another logic control device, a program that may reside on this new startup device, and/or a logic control device, and/or StorExecute, may execute and display a list of various options, for example, methods to "fix" the data storage device that is in need of repair, and/or update Master Template.

An optional program and/or script may also execute (or give the user a dialog box offering the user to select) functions to be performed such as: user may choose to reformat the data storage device that needs repair ... and/or the user may choose to execute a low level or quick reformat of the data storage device, or the user may choose to skip the reformatting process. When choice is selected, the function is performed by a program that executes that function, e.g., reformatting software is run.

Optionally, the user may also be asked if he/she wants to copy all or some of the data that exists on the Master Template data storage device(s), and/or Backup data storage device(s), and/or Archive data storage device(s), to the data storage device(s) that need(s) to be repaired. If this choice is selected, then the same type of program that was used to make the backup(s) could also be used to copy and/or modify the data from the Master Template back to the user's data storage device, thus making the data storage device used by the user (nearly) identical to the Master Template. (There may be some variations made to the data on the repaired data storage device as needed-- for example, shortcuts and/or aliases may need to be modified to point to their targets correctly).

Optionally, a program and/or script may also execute (or give the user a dialog box asking the user to select) any of the following functions to be performed: copy data from one data storage device (from a particular date or partition) to a different data storage device, as needed. For example, the same type of program that was used to copy User Documents, Email, Bookmarks/Favorites, Quicken data and/or preferences, etc. to the Backup Storage Device, can now reverse its process and copy all or some of the data back to the user's data storage device that needed repair. (A script may be executed that gives the user the choice of which backup (e.g. from which date) to copy the data).

FIG._17X

Optionally, a program and/or script may also execute (or give the user a dialog box asking the user to select) any of the following functions to be performed: modify data, and/or perform ANY instructions ("de-fragment", virus scan, re-partition, etc.) that affect ANY data storage device that is in need of repair and is accessible/available to the computer system locally or across a network.

Upon successful completion of the repair process, a program and/or script may execute that causes the Switching Process to select the original "startup device," and the repaired data storage device can continue to operate normally.

The Master Template System and/or Backup System, and/or original data storage device may continue to be utilized to perform their functions.

The "Backup and/or Repair" system Switching System is utilized as described in section 7. Optional scripts/programs designed for various other functions can be combined as well.

9.2.2 In reference to figure TW 169:

This is another example of one method of an optional aŭtomatic repair script/program:

Upon computer startup, a script/program can hide all activity that occurs in the background. During this time, a logo and text or other information and/or graphics may be shown to user. This text and logo could be able to be modified by the user. Optionally, this "hiding" function can be toggled off/on at any time by keyboard input of a password and/or use of (a) "hot" key(s) by the user.

Optionally, a program or script could run the following sequence of events (all in the background hidden from the user):

A backup and/or archive program could be run that makes a complete backup and/or archive of the data on the data storage device that is normally used (e.g. at device ID 1). The destination of this backup and/or archive could be able to be modified in user preferences. For example the backup/archive could go in a partition or folder on the drive at ID 0, or could go on a drive at ID 2.

Optionally, a program or script executes that checks to see that the backup has been made successfully. After confirmation that backup is complete and successful, user can be given an optional dialog (that can be modified by client) allowing user to select one of the following options: (ID 1 is used as an example) no format of ID 1, quick format of ID1, low level format of ID1. Optionally including complete deletion and recreation and/or re-write of all partitions, master boot records, etc. (Background could remain hidden) Optionally, this dialog only shows up if selected in preferences. In lieu of this dialog, preference can be selected by the user as to which type of format is performed or not performed.

Optionally, if user is given the option in "user-preferences," based on user selection, quick, or low level, or no format is run on ID1. Otherwise whatever default option is in preferences is done. Optional script then executes a program that copies some (not all) data on ID 0 to wherever data belongs on ID 1. For example this script may copy such items as: Explorer Favorites, Netscape Bookmarks, E-mail, in box, out box, and address book. Optional: LED and/or computer screen can provide a dialog box that says something like: please turn switch to "Normal Use"

FIG._17Y

position. Optional: LED and/or computer screen can provide a dialog box that says something like: "please restart computer". Or those events can happen automatically, with or without dialog. Optionally, all the events described above can also be written into a StorExecute.

9.2.3 Another example of controlling the repair process. See figure TW169

• An optional time delay circuit can be integrated that allows the PRAM/CMOS/CUDA, etc. to be reset prior to performing the rest of the repair process.

• Optionally, a controller can reset PRAM/CMOS/CUDA/BIOS, etc.

• Optionally, a logic control device can delay switching between data storage devices until restart, and/or can avoid restart by use of switching control to another logic control device, and/or can conduct repairs without restarting.

• Optionally, a second processor and OS can run in the background so that switching for repairs does not require restarting.

Please Note:

In all wiring and circuit board diagrams, only the material are illustrated, and it will be understood that they are not drawn to scale.

9.2.4 The "Backup and/or Repair" system as used for Repairing Several/Many Data Storage Devices, Multi User Repair:

The repair process can be used to repair multi-user computing devices, and/or device with multiple data storage devices using the same techniques as described herein for single data storage devices.

9.2.5 The "Backup and/or Repair" system as used for Emergency Startup - How it Works:

The "Backup and/or Repair" system can be used for switching to an emergency startup system and/or device. When a startup system and/or device can't be seen, and/or at the request of the user, and/or other request from elsewhere, the "Backup and/or Repair" system can switch to a secondary startup system and/or device, and/or to switch between RAID mirror data storage devices and/or systems. The "Backup and/or Repair" system Switching System is utilized as described in section 7. Optional scripts/programs designed for various other functions can be combined as well.

9.2.6 The "Backup and/or Repair" system as used for formatting and device testing - How it Works:

The "Backup and/or Repair" system can be used to test data storage devices. It can be used to temporarily switch to a data storage device for startup, while a data storage device reformats and/or tests another different data storage device. It does this by switching from the data storage devices in-use, to alternate logic control and/or data storage devices that are used while testing and formatting takes place.

FIG._17Z

9.3 The "Multi-User System" as Used for Multiple Operating Systems and/or Software - How it Works:

The "Multi-User System" can be used to have totally different setups of operating systems and software, and switch back and forth between them. So for example this would enable a computer to be set up with Linux and movie editing software in Japanese, and the "Multi-User System" could then switch the computing hardware to be set up with Windows and mathematics software in German.

9.3.1 The "Multi-User System" as Used for Multiple-Users - How it Works:

The "Multi-User System" can also be used to enable multiple users to use computing hardware as if each of the users had their own private computer. When a user is using a computer, the "Multi-User System" sets up the data storage device, operating system, applications, etc. just for that particular user. Then, when a different user wants to use the computer, the "Multi-User System" hides away the previous users data storage device, operating system, software, and data, and provides a different data storage device, operating system, and software for the new user.

9.3.2 The "Multi-User System" as used for Switching Between Several/Many Data Storage Devices - How it Works:

Data Storage Device Switching: "Multi-User System" can also be used for rapidly switching between many different data storage devices such as those used on computers. Please note that if the data storage devices are set up as bootable startup devices, then the "Multi-User System" can switch rapidly between startup data storage devices.

9.4 The "Virus-Proof/Hacker-Proof System" - How it Works:

A "Network accessible" data storage device defined:

A network accessible data storage device could usually be switched partially or completely off, and/or the network connection could be switched off, and/or the network accessible data storage device could be "un-mounted." The network accessible data storage device could: 1) only be mounted and/or connected to the network, and/or turned on, when used for sending and/or receiving data on the network; or 2) could always be network accessible; or 3) sometimes be network accessible.

Optionally, the network accessible data storage device could be limited to containing only non-sensitive software, and/or outgoing data waiting to be uploaded or sent.

Optionally programs could exist on the network accessible data storage device that enables mail to be sent and/or received, but not opened.

The program could: unlock and/or open network connections, send and/or receive mail, upload and/or download data, close network connections, mount and or turn on, and/or connect to another data storage device, send downloads and/or mail received to another data storage device.

Quarantine data storage device

FIG._17Z-1

A Quarantine data storage device is utilized in the transfer of data back and forth between isolated segments of a computing device. Data may be copied from the "Internet Accessible" and/or "Temporary" data storage device to and from the Quarantine data storage device and then into the protected user data storage device as needed to isolate/protect data. Optionally, after being held in quarantine for a period of time, data can be checked with the latest virus checker before being copied to the protected user data storage device.

Additional information

If any computer on the network detects a virus and/or starts missing data, it notifies all nodes on the network and a preset protocol of response takes place that may consist of one or more of the following: all nodes on the network make backups; all nodes download most current virus checker; all nodes check themselves; backup schedule for all nodes may increase, etc.

Multiple communication cards with various identities may be used to switch identities and send/receive data. A computer send/receive function may be set up like a shell game, where the identity is changing rapidly, and the data storage device is on-line for as short a time as possible... just long enough to send and/or receive, and then it is taken off line. Rather than using a separate drive for viewing data, software would not allow viewing until drive was off line... then before going back on line, software would transfer all data except outgoing data to quarantine.

A program may also synchronize (or copy on a time-delayed schedule) emails (and or other files) that were sent and/or received from the "Temporary" data storage device to the "Quarantined" data storage device, as needed.

(Optionally, data on a data storage device (and associated computing hardware that determines date/time) may be "duplicated" onto a different data storage device with one significant difference: it could be set to use an "older" date/time setting so that if a "date-triggered" virus is present it would not trigger on the "older date" data storage device.)

9.5 The "Hardware-Repair System" as used for Emergency Computing - How it Works:

The "Hardware-Repair System" can be used for switching to emergency backup computing when a computer fails due to hardware problems. The "Hardware-Repair System" can switch to a secondary computing device, and/or to switch between computing components

9.5.1 The "Hardware-Repair System" as used for formatting and device testing - How it Works:

The "Hardware-Repair System" can be used to test computer hardware components, circuit boards and computers. It can be used to temporarily switch to a data storage device for startup, and then tests hardware components. It can be used to switch to a different logic board, network connection, or computing system, while it tests the components that were taken out of use. It does this by switching from the devices in-use, to alternate devices that are used while testing and formatting takes place.

9.5.2 Software Switch Replacement - How it Works:

A software switch replacement can use a program that performs many of the same functions as a switch.

FIG._17Z-2

For example, imagine two computers: Computer A and Computer B. A hardware switch could be used to turn off Computer A, turn on Computer B, and switch jumper(s) on a mirror data storage device on B to ID 0.

Software can do the same thing. For example, a simple program in ROM or elsewhere can be used to tell the computer (or computing device) to switch computing devices and use a mirror of the data storage device.

Software can select which computer circuit boards, devices, components, and hardware to utilize.

9.6 The "Freeze-Buster System" - How it Works:

The "Freeze-Buster System" includes a switching process that can switch and reset connections to devices. To do this it switches on/off one or more wires that run between the device and the computer/computing device. It can be used to switch any connections such as data, and/or power, and/or ground. It can switch one or more devices. It can also switch indicators (such as LED lights) for activity, power, and identity. As needed, it can utilize optional scripts and programs to enable use of these new switching abilities. Another option, it can switch local computing hardware, and/or switch over a network.

For example, imagine that an external "hot-swappable" data storage device is attached to a computer. "Freeze-Buster System" can be integrated into the device, or put inline, or on the computing device. It switches the connection to the device off/on as needed to break out of a freeze, and/or reset the connection.

Freeze-Buster can be used to reset connections with external and internal devices.

9.7 The "Net-Lock System" - How it Works:

The "Net-Lock System" includes a switching process that can switch one or more of the following: network and communication connections. It can switch indicators (such as LED lights) for activity, power, and device identity settings. As needed, it can utilize optional scripts and programs to enable use of these new switching abilities. Another option, it can switch local computing hardware, and/or switch over a network, and/or global computer network/global communications network. (for example see figures Fig._26A - Fig._26J, Fig._24E.

A program and/or script may also execute (or give the user a dialog box asking the user to select) functions, for example, but not limited to:

- Lock/Unlock Network and/or communications connection.
- Setup remote lock and unlock preferences
- Schedule lock and unlock times
- Create or modify list of users who are authorized with lock/unlock privileges

9.7.1 Network Privacy and Security - How It Works:

The "Net-Lock System" can also switch on and off network connections. It can use the "Net-Lock" switching system to turn off and on (and lock and prevent access to) connections to a global computer/communications network, intranet connections, and all other types of network connections.

FIG._17Z-3

9.8 The Entertainment Center System as Used for Computer Repair - How it Works:

The Entertainment Center consists of electrical connections, holders, fittings, etc. on the inside, outside, and/or integrated into the body of a computing device that provide the ability for the user (or manufacturer) to hook up anything they want that may interact with the computer and provide entertainment, education, artistic value, etc. For example the outside of a computer can be covered in part, or completely with electrical connections that allow a user to attach devices. Examples:

Example 1: Flashing lights plug into a device that is attached or is itself the computing case. User can change where lights plug in. Lights interact with all switching events and/or with the user(s)

Example 2: A model and/or robot (or robotic device) modeled/constructed in the likeness of a person and/or animal and/or creature and/or thing and/or device, with computer built in Example: Robot that appears human, holding flat screen monitor, sits cross legged on users desk. Wireless keyboard option. Speech, movement, vision, etc. can be integrated into this robotic device. As part of the interface with the computer, the robot can explain what is happening, and discuss events taking place on the computer, work performed by the user, daily news, information gleaned from the internet, ask questions about how the user is feeling, etc.

Example 3: It can be used for art projects, education, entertainment, can consist of any creative use a user wants to make of it. For example, it can be an aquarium, or a hamster house, (in which case attachments can monitor and interact with the pet, and give user feedback. (i.e.: number of cycles a hamster has run on a wheel, etc.) It can be used for educational projects, anything at all. The idea boils down to this: A computer/computing device, does not need to look like a computer. It can have electrical connections and methods of attaching things that enable a user to do anything creative and constructive with the outside of the computing device. Events taking place in the computer can interact with components attached to the computer, and/or in communication with the computer. Components attached to the computer, and/or in communication with the computer can interact with the computer and information can be exchanged, processed, controlled. The computer can control the attachments, and/or the attachments can interact with the computer.

Example 4: The entertainment center can have switches that turn things off/on and interact with the computer

Example 5: The entertainment center, that is connected to and/or houses the computer/computing device, can look like anything. Whatever someone wants to create. And any artistic endeavor.

Example 6: Entertainment center: Can use interactive modular devices, modular components that fit together, and/or non-modular components.

Example 7: The entertainment center can utilize and/or interact with any form of modular and non-modular devices that can plug into it. People can create whatever fantasy they like. Any sort of modular and non-modular kits can be built to be utilized with the entertainment center and for any purpose such as education, entertainment, games, science, robotics, chemistry, art, lighting, whatever.

Example 8: The entertainment center can run and/or be integrated with a biosphere, terrarium,

FIG._17Z-4

garden, food dispenser and/or diet control device that may provide food on a schedule.

9.9 Method of Creating Custom Cables and Connections

Figure Fig._22Z-1 description

A method of constructing new types of cables and/or connectors to enable switching.

"A" is a cable utilizing a new type of connector "B". The view of B is a top or bottom view. In this example 3 of the wires need to be switched, so the cable has been manufactured with 3 of the connections split. For example connection "N" is connected on one side to the cable, and the other side is plugged into a hard drive, but the connection between the two sides leads to separate pin outs on the cable, (in this example they are located at the top or bottom of the cable). These pin outs can then be connected to wires or a cable that is then attached to a switching device with switches the connection(s).

Example J below is similar to the connector above, but it is not built into a cable. Two cables could be plugged into each end of it, or a cable could be plugged in one end, and the other end could be plugged into a data storage device for example.

Example K is a top or bottom view of a connector in which each of the wires is switchable, and diagrams L and M shows side views of the same connector. The connections in these special cables and/or connectors can be male and/or female.

The switching process can be controlled manually, and/or by a logic control device.

F shows a switch with optional logic controller which can be plugged into standard cables and connectors, and that can switch one or more of the connections.

Please note that these new type of connectors don't need to have the new pin or connectors at the top or bottom, they can be located anywhere convenient. They can be any shape and/or side, and be designed for any type of cable.

Alternatively, the wires themselves can be routed directly to a switch (and back) as needed. Thus, cable A can have one or more of the wires routed directly to the switch as wires and/or as a cable.

This can be used to switch any type of cable, for example, it is possible to switch device ID, read/write, power, perform cable select switching, switch hardware devices, individual components, etc.

Figure Fig._22Z-3 description

Figure Fig._22Z-3: The one or more wires in a cable can be cut and go to a switch. This can be used to switch any type of cable, for example, it is possible to switch device ID, read/write, power, perform cable select switching, switch hardware devices, individual components, etc.

9.10 The inventions described herein can be mixed and matched as needed

FIG._17Z-5

9.11 Anti-theft system: The Anti theft system can contain one or more of the following: cellular phone technology, a global positioning system a transmitter/receiver, a meaning of identifying the user, and an extra data storage device, logic control, and a switching process. Using all or some of these devices it can use any means of identification to identify user. If user does not match authorized user the following events can occur:
1) User data storage device is switched off and is thus "hidden"

2) A "bogus," but normal looking data storage device is switched "on," and mounts (optionally id may be switched). It may optionally have a hidden partition that is protected from being erased. Software executes that may be hidden and/or misnames, and/or otherwise would not draw attention from the user and sends out machine location to for example, police, owner's e-mail address, a security service. Information can also be transmitted using any type of transmitter, example: cellular phone call, and/or be sent over a network and/or the internet. The anti-theft process could also be triggered by a phone call.

If user identity doesn't match authorized user, the device may hide the user data storage device, switches to "rigged" data storage device, and also may turn on a global positioning system transmitter to identify location of computer.

When the location of the computer is identified, it can be tracked, even if it is moving.

9.12 A combined repair system:

The hardware repair system can be combined with the other parts of the invention to provide a unique combination of repair functions. For example, if the hardware repair system included all of the features described herein, the combined inventions would provide a computer that could repair any software and hardware problem, be immune to hackers, be extremely virus proof, provide entertainment, interface, artistic, and educational features, etc.

In one example, the Self Repairing System utilizes two processors, and can use two discrete computing systems (with an optional shared data storage device, and/or mirrored data storage devices and/or quarantine data storage device(s) that may be integrated into one box) and can be combined with the hardware repair system.

Combined hardware & software repair example: If a second computing process is utilized, repairs can happen on the fly, perhaps without notice by the user, and/or with little interruption to the user. An integrated secondary backup computing system can always be "up and running", with the ability to automatically detect a problem, switch to the backup system, and conduct software repairs on the fly, and/or conduct repairs from the backup system, without bothering to switch the user to a secondary system (unless needed) because the repair process can be so fast the user may not even notice a problem. For example, a freeze occurs. The secondary system can detect the freeze, reset connections, replace defective software components, and clear and reset devices as needed, so fast that the user may never notice that a freeze took place

The repair process can (optionally) utilize a comparative process that compares the software on the user data storage device to a perfect Master Template. By monitoring user processes, we can monitor the state of the user template, be aware of changes (optionally, have a database of changes and/or differences between the user data storage device and the master data storage device) and rapidly repair the user data storage device (on the fly) as needed based on that database, reset

connections, and reset memory if needed.

As an option, system does not need to discard (and/or overwrite) user documents, email, etc. so if there is a freeze or corruption problem only the system software, and/or offending software that is having a problem can be repaired, and just those components that are different from the master can be replaced.

9.13 Optionally, the user operating system and/or applications, and/or data and the Master Template, and repair process can be run in volatile memory, enabling a fast repair process to perform much more quickly, especially if a comparative repair process is used that repairs problems as they occur. Thus, if a "process watcher" is used to detect a problem, the fast repair process can happen so quickly it may not even be noticed by the user.

9.14 Another version of the web site repair process is to use the repair process to repair the web site on a continuous basis during use, and/or optionally switch to other web servers during the repair sequence. Thus, a web site can always be kept in "perfect" condition. If the anti-virus/anti-hacking system is integrated, it could also prevent hacking past the web site. In this case, for example the web site could reside on the data storage device accessible to the network and/or internet, but all other data on the web server would be isolated, not connected to the network, and therefore "un-hackable."

Optionally, an integrated logic control device can reject additional users and close the site down for repairs when defects in the site are detected (during the process of comparison with the master template). To do this, during the repair process, when it is noticed that files and/or data is changed, modified, and/or different, the software then sends a command to the logic controller to reject further hits to the web site, (and optionally disconnect web site users), and conduct repairs as needed.

9.15 All of the inventions herein can be integrated into any device that utilizes a computer and/or computing device, such as televisions, radios, network appliances, machinery, vehicles, etc.

9.16 Net-lock and Freeze-buster can be internal and/or external devices. Either of these devices could lock into place using any sort of locking system and/or holder and/or retaining mechanism and/or device.

9.17 In Anti-virus/Anti-hacker can switch back and forth between a "side" of a computer connected to the internet, and a "side" that is isolated from the internet, by use of a hardware and/or software switching process.

9.18 Dual network connection cards (and/or multiple "cards") can be controlled by a single switch and/or switching process.

9.19 Net-lock can be triggered by software and can be a software switch, without use of hardware.

FIG._17Z-7

Fig._20 DIAGRAMS

The diagrams that begin with "Fig._20" represent the concepts of what we are switching, and that power, ground wire(s), device ID(s), startup order, data, etc. may be switched, and may bypass the switch as needed. They represent circuitry that could optionally bypass the switching mechanism; for example: power and/or ground and/or data (including via SCSI, Firewire, USB, IDE, and all other types of data communication). We assume someone skilled in the arts can use a bit of common sense here.

Please note that data storage devices, switch triggers, etc. in diagrams can be "local", or utilized over a network.

In the Fig._20A-20H, the circle with an X in it represents the switching process. The lines to the circle with the x show what is switched, and lines that go around the outside of the devices, (and not to the switch) represents the concept that the data does not need to get switched (but it can be switched).

Any combination of the 5 V and/or 12 V and/or ground and/or other power may be switched, either individually, or in combination. It is possible to switch the jumper cable connections that determine the boot order upon system startup. These connections are indicated on Fig._20A by the letters (a) (b) and (c).

Also, ground wires have not been shown, because anyone skilled in the arts can hook up ground wires (or interrupt them).

Data storage devices shown in these drawings can be hot swappable, "local", or located on a network.

When computing equipment is used with any other type of device(s) (for example: robots, robotics, transports systems, televisions, telecommunications, manufacturing, equipment control, etc.) the Switching System switching system and/or switch trigger can be relocated, and/or additional switches and/or switch triggers added, so as to easily provide accessibility users and/or people controlling the system. Additionally the switch mechanism and/or trigger can be camouflaged or hidden as needed. Location appearance, and type of switch can be changed as needed.

Fig._20A: Example of Backup and Repair System: the Switching System (S1500) is used for repair of one or many data storage device(s) (S1150) switched via connection (b), and connected to computing system hardware (S1100) via connection (f). Master(s) and backup(s) reside on one or many data storage device(s) (S1700) connected to the computing system hardware via (g) and switched via connection (a). Electrical power, and/or ground wire(s) and/or jumper cable connections, and/or any connection that determines the boot order upon system startup, may all be independently or collectively switched by the switching process. These various connection types are indicated on Fig._20A by the letters (a), (b) and (c) and can be switched by the switching system (S1500). In the event that ID switching is used, the ID of the startup device can be switched to a different ID, and another data storage device can be switched to the default startup device position.

Optional PRAM, CMOS, CUDA, BIOS, EPROM (or other memory storage module and/or device) reset is indicated in box (9202) and controlled by the switching system (S1500 via the connection

(d) and powered by connection (e).

Fig._20B: Example of Backup and Repair system Switching Process as used for repair of one data storage device. Master and backup on separate data storage devices, with archive.
Fig._20C: Example of Backup and Repair system Switching Process as used for repair of two or more data storage devices. Optional startup device.
Fig._20D: Example of Backup and Repair system Switching Process as used for repair with other processes such as switching circuit boards, chips, devices, device identity, data storage devices, circuitry, global positioning transceiver & transmitter anti-theft and positioning system, computing hardware systems, ROM, backup storage devices, identity indicators, remote trigger and/or switch, and any StorExecute (see definition).

Fig._20E: Example of Multi User System as used for switching between multiple operating systems, software setups, storage devices, and/or circuitry, and or templates.

Fig._20F: Example of Backup and Repair Switching System as it is used for repairing multi-user and/or multi-use use, (such as parents and children.)

Fig._20G: Example of Backup and Repair process as used with many data storage devices and/or startup devices and/or circuit boards.

Fig._20H: Example of Backup and Repair Switching System as used for emergency startup and/or operation.

Fig._20I: Example of Hardware-Repair Switching System as used for switching between computing devices. Thus, if one computing device fails, it is simple to switch to a secondary computing device. S9110 optional combined hardware represents the concept that such items at keyboard, mouse, monitor, etc. can be shared, or seperate.

Fig._20J: Example of BAR system Switching System as used for switching data storage device identity. A relay and/or switch can be used to break and connect jumpers as needed to change device identity.

Fig._20K Switching process with dual computing devices and ability to switch data storage devices.

Fig._20L Examples of Switching System triggering methods: switch trigger (a) triggers the Switching Process (b).

Fig._20M Switching process for the purpose of isolating data so as to protect data from malicious code and/or "hackers" that are "snooping" via a network connection.

Fig._20N Switching Process switching computing hardware devices in order to isolate and protect data and to include the repair ability of Fig._20A.

Fig._20O Switching Process controlling dual/isolated computing systems for the purpose of isolating data (so as to protect data from malicious code/un-authorized network "prying eyes" ("hackers).

Fig._20P Switching Process switching computing hardware devices in order to "repair" the computing device, e.g. the ability to switch from a "failed" device to a "backup" or "redundant" and

FIG._18B

"functioning" device.

Fig._20Q Example of switching a network connection "off" or "on" and then back again.

Fig._20R Example of the ability to interrupt a connection to a connected computing device for the purpose of "resetting" or "unfreezing" a connection and/or device.

Fig._20S Example of isolating/protecting data by switching data storage devices and network connections.

Fig._20T: Example of various "systems" capable of being integrated with one another in a interconnected (optionally modular) manner. Note: each "system" can be "mixed and matched" selectively with other "system(s)" to function independently and/or dependently.

Fig._21 DIAGRAMS

The diagrams that begin with "Fig._21" are flowcharts.

Fig._21A: This example shows the switching process between data storage devices. The switching process switches to a different startup device and/or group of data storage devices.

For example: User is using data storage device "A", then triggers the switching process, and device "A" is deactivated, and device "B" becomes the new startup device.

"A" and "B" can each represent one data storage device, or an entire group of data storage devices. Although the diagram only shows switching between A – G, there is actually no limit to the number of devices it can switch. Please note that: a user, schedule, or event may trigger the switching process.

Fig._21B-Fig._21O: Diagrams Fig._21B through Fig._21O are considered to be one continuous flowchart extending over seperate pages.

Fig._21B: Example of sequence of events including optional backup, archive, Master template creation/update.

Fig._21C: Example of sequence of events including: switching startup devices; reset of PRAM/CMOS/CUDA, BIOS, EPROM (or other memory storage module and/or device) etc.

Fig._21D: Example of sequence of events including options offered to user.

Fig._21E: Example of sequence of events including more options offered to user.

Fig._21F: Example of sequence of events including method of virus checking.

Fig._21G: Example of sequence of events including backup and archive of data.

Fig._21H: Example of sequence of events including options for user to run various types of repairs.

Fig._21I: Example of sequence of events including options for user to select startup device.

Fig._21J: Example of sequence of events including options for user to select automatic repair preferences.

Fig._21K: Example of sequence of events including options for user to select type of data storage device format.

Fig._21L: Example of sequence of events including an example of repairs performed based on preferences.

Fig._21M: Example of sequence of events including backup/archive and options for user to select which backup or archive to "revert to".

Fig._21N: Example of sequence of events including update of master template.

Fig._21O: Example of sequence of events including options for user to revert to prior bookmarks, E-mail, and other items.

Fig._21P-1 & Fig._ 21P-2: Switching Process occurring without shutting the computing device down and restarting by utilizing two or more processors (and/or "protected processing") and two or more memory devices (and/or "protected/segmented" memory).

Fig._21Q: By isolating data, potentially malicious code (e.g. viruses) cannot destroy other isolated data because it does not have access to the "protected" data.

Fig._21R: The Switching Process (5210) isolates data (5220) that exists on or in computing devices.

Fig._21S Examples of the types of devices where data can exist (and thus isolated).

Fig._21T-1 & Fig._21T-2 Example of booting into different data storage devices for the purpose of isolating data on different data storage devices.

Fig._21U Example of isolating data in volatile memory by "flushing" volatile memory, or "emptying" its memory, before and after switching to different data storage devices.

Fig._21V Example of isolating data in volatile memory by utilizing two discreet volatile memory devices when switching between different data storage devices.

Fig._21W Example of isolating data in volatile memory by switching between discreet "segments" of a single volatile memory device when switching between different data storage devices.

Fig._21X Example of isolating data in processors by utilizing two discreet processor devices when switching between different data storage devices.

Fig._21Y Example of isolating data in a processor by switching between discreet "segments" (or "protected" addresses) of a single processor device when switching between different data storage devices.

Fig._21Z Example of the Switching Process (6110) occurring so as to physically switch "on" or "off" (by "breaking" or connecting the circuit/wiring) power connections (6120) and/or data connections (6130) of a network connection device (6140) as utilized by a computing hardware device (6150).

FIG._18D

Fig._21Z-1 Example of the Switching Process (6210) occurring so as to "logically" switch data connections of a network connection device (6240) by ignoring data (6220) or by ceasing to process the data (6230). A computing hardware device (6250) typically utilizes network connection devices.

Fig._21Z-2 Example of a "brief" Switching Process (6310) occurring so as to physically switch, momentarily, from "on" to "off" and then quickly back to "on" (by "breaking" or connecting the circuit/wiring) power connections (6320) and/or data connections (6330) of a peripheral computing device (6340) as utilized by a computing hardware device (6350), for the purpose of "resetting" the device.

Fig._21Z-3: Example of switching system connected to a computer that is connected to a network (e.g. a web server connected to the Internet) and many data storage devices (a, b, c, d, e, f -- could be many more). Switching system "cycles" between data storage device so that only one data storage device is utilized (online and connected to the network) by the web server at any given moment. Note: data storage devices may also be "repaired" (see Fig._20A) by repair system (while not connected to the web server) if they become corrupted or "hacked" (defaced).

Fig._21Z-4a & Fig._21Z-4b: Example of a Switching System that "records" data from one memory device to a second memory device so that if memory in first memory device is deemed to cause undesirable events in the processor, the user can "revert" to the code/data that was executed before the undesired event occurred, thus allowing computer (and/or user) to avoid the activity that caused the problem the first time.

Fig._21Z-5: Example of Switching System used as an Anti-Theft device to protect a user's private data. Optionally, it may verify ID as method to trigger Switching System. Optionally, system may send a signal to identify the computer's location.

Fig._21Z-6: One example of the direction that data is allowed to "flow" from one device to another.

The diagrams (Examples: Fig._21Z-7-Fig._21Z-9, Fig._21Z-10, Fig._21Z-6) are meant to represent the concepts of what we are switching, and that power, device ID, network connection, data, etc. may be switched, and may bypass the switch as needed. They represent circuitry that bypasses the switching mechanism such as power when it is not switched, ground, data (including via SCSI, Firewire, USB, IDE, and all other types of data communication). We assume someone skilled in the arts can use a bit of common sense here.

Fig._21Z-7-Fig._21Z-9: Flow chart of step-by-step scenario for process of protecting data from virus and hacker attacks.

Fig._21Z-10: Devices the computing device switches.

Fig._22 DIAGRAMS

Wiring Examples:

Circuit Boards: Figures starting with the letter "W" show relays and wires which we used for building prototypes. We have not duplicated the drawings into their circuit board versions because anyone skilled in the art can perform the same functions via circuit boards.

Please note that data storage devices, switch triggers, etc. in diagrams can be "local", or utilized over a network.

We have used relays and wires in the drawings, but assume that anyone skilled in the arts can convert relays and wires to circuit boards.

Data storage devices shown in these drawings can be hot swappable, "local", or located on a network.

When computing equipment is used with any other type of device(s) (for example: robots, robotics, transports systems, televisions, telecommunications, manufacturing, equipment control, etc.) the switching system and/or switch trigger can be relocated, and/or additional switches and/or switch triggers added, so as to easily provide accessibility users and/or people controlling the system. Additionally, the switch mechanism and/or trigger can be camouflaged or hidden as needed. Location appearance, and type of switch can be changed as needed.

In most of these diagrams the 5 V and 12 V power can optionally be switched, but in most cases both can be switched, but it is only necessary to switch one or the other.

Also, ground wires have not been shown, because anyone skilled in the arts can hook up ground wires.

Fig._22A: Example of Switching Process switching the data storage devices for the purpose of repair. This is accomplished by switching power and jumper Ids to change the startup order of said data storage devices.

Fig._22B: Multi-User computing system, with data storage device identity switching, PRAM, CMOS, CUDA, etc. reset, Switching System repair process, and use of brake, clutch, circuit board, programmable logic controller, etc.

At W5110 the circuit board/PLC can time delay the sequence of events so that the reset relay (for example see W5117) is triggered first, and then after completion of reset, the rest of the switching process continues. Thus, if PRAM, CMOS, etc. is corrupt or set for a different startup device, it will be cleared prior to the next steps. Also, optionally after reset, a program or script can be run that sets the startup sequence and/or resets PRAM/CMOS, etc. to predetermined settings.

Fig._22C: Single User with repair.

W53: Repair and backup of Multi-User and/or multi-use system, and PLC and/or circuit board, and reset of PRAM, CMOS, CUDA, etc. Also, example of modular approach to backup and repair process. The same wiring and circuitry methods can be duplicated over and over for more users/more devices.

Fig._22E: Repair and backup of Multiple Data Storage Devices. Shows repair, backup, and individual switching of power and device identity. Please note that the device identity switch could be the type that is usually on the back of external SCSI devices that can be switched from 0 -6 or the type that can be switched from 0 - 15, or other switches that can perform switching to multiple identities. Also, example of modular approach to backup and repair process. The same wiring and circuitry methods can be duplicated over and over for more users/more devices.

FIG._18F

Fig._22F: Repair and backup of Multiple Data Storage Devices. Shows repair, backup, and individual switching of power. Please note that device identity is not switched in this version. A version can also be constructed that has some of the devices switched, and some not switched. Also, it is an example of a modular approach to the backup and repair system. The same wiring and circuitry methods can be duplicated over and over for more users/more devices.

Fig._22G: Repair of Multiple Data Storage Devices. Repair on separate busses.

Fig._22H Repair of Multiple Data Storage Devices utilizing hot-swap drives.

Fig._22I: Switch used for switching computers, computing devices, computing hardware. Thus, if one hardware device fails, just switch to second device. Please assume that switch W5911 can utilize the brake, PLC, circuit board controls shown in many of the other figures.

Optionally do not switch ground. Optionally, isolate ground from other computing device(s).

Fig._22J: Single user, with repair, with switch to secondary computing device and common mirror. Continued on Fig:_22K.

It should be mentioned that on diagrams 60 and 61 and other diagrams that use shared computing hardware, that optionally ground wires can isolated, and/or switched as needed to isolate each of the computing devices. Surge and voltage protection and filtering can also be added between the data storage devices and computing devices.

W6060 can be a switch or a relay. In diagrams Fig._22J and Fig._22K, instead of using two relays for the master, one relay can be used if the wires from CPU A and B are isolated power coming from the power supplies. For example when CPU "A" is on it won't effect CPU B if B is isolated. The device can also be constructed without isolating the power, and without any switch or relay although that construction method is not recommended.

Fig._22K: Single user, with repair, with switch to secondary computing device and common data storage device mirror. Continued on Fig._22J.

Fig._22L: One computer and/or computing device containing two computers and/or computing devices with mirror and ability two switch between devices. The computing devices can be set up with multi-user, repair, etc., but with common mirrors and ability to switch between computing devices. This system can also be built as separate units instead of combined in one box.

Fig._22M represents examples of wiring diagrams for interrupting connections to computing devices. The diagram shows four examples of this: A, B, C, and D. This is a functional diagram only. Additional wires can be added and switched as needed. Interrupting connections can also apply to switching wireless connections. The connection is briefly interrupted for the purpose of "resetting" the device.

Fig._22N represents examples of wiring diagrams for turning network connections "off" and "on". The diagram shows four examples of switching a network connection: A, B, C, and D. This is a functional diagram only. Additional wires can be added and switched as needed. Can also switch wireless connections.

FIG._18G

Fig._22O Example of interrupting connections to an external device to reset connection and/or for the purpose of resetting a device, connection, or to "break" out of a "freeze."

Fig._22P Example of computing device containing dual computing devices that can be switched, plus a shared data storage device (that can be switched back and forth between the dual computing setup) for the purpose of isolating data (so that malicious code cannot affect other data).

Fig._22Q: Repair of a Multi-User System.
Assumptions about circuit board:
If there is no power to circuit board when computer is shut down:
Circuit board won't operate and can't break CMOS circuit. In this situation it is best to just bypass circuit board and only break CMOS circuit when key is in momentary position.
If there is power to circuit board when computer is shut down AND if there is no power to logic board in shutdown mode:
Circuit board can be used to break power to CMOS prior to computer startup.
If there is power to circuit board when computer is shut down AND if there is power to logic board in shutdown mode:
Circuit board can NOT be used to break power to CMOS prior to computer startup.
Circuit board can be used to send "zap PRAM" keyboard sequence to logic board on startup.

Fig._22R Similar to Fig._22P but also shows how data can be further isolated: a network connection can be switched to ensure isolation of data. For example, the network connection can be switched "off" whenever data storage device (6215) is "on" and the network connection can be switched "on" when data storage device (6214) is "on" (and vice-versa). This "Virus-Proof/Hacker-Proof" computer is a computer that uses one (or more) data storage devices for normal use, and a different data storage device(s) for doing E-mail. The Switching System switches between the data storage devices, alternating between "active" and "inactive" data storage devices. To move data from the E-mail data storage device to the hard drive, or visa versa, a temporary "quarantine" data storage device is used. Optionally, it will not release data until an on-line connection has been made and the drive checked with a current virus checker. Data can optionally be held for a time period, and then released upon a virus check.... giving data virus companies time to detect new viruses and update their software.

Software can be used to replace the Switching System switch or in conjunction with the Switching System switch.

Optionally, the Switching System switch can leave a network connection "on", and switch "off/on" a separate connection to the internet/global computer and communications network. Or both the network connection and connection to the Internet can be switched separately

Fig._22S Circuit Board and Socket Assembly Option

Fig._22T: One type of circuit board for Repair and Backup

1) circuit 1
2) circuit 2
3) circuit 3
4) circuit 4
5) circuit 5
6) circuit 6

FIG._18H

7) circuit 7
8) circuit 8
9) circuit 9
10) circuit 10
11) circuit 11
12) circuit 12
13) circuit 13
14) circuit 14
15) circuit 15
16) circuit 16
17) circuit 17
18) circuit 18
19) circuit 19
20) circuit 20
21) Power Control Indicator #21
22) Power Control Indicator #22
23) Power Control Indicator #23
24) Power Control Indicator #24
25) Power Control Indicator #25
31) Power Control Indicator activity light for #21
32) Power Control Indicator activity light for #22
33) Power Control Indicator activity light for #23
34) Power Control Indicator activity light for #24
35) Power Control Indicator activity light for #25
26) time delay circuit
27) Data and power to LCD screen
and/or data for computer monitor
and/or to computer.
28) Power to board
50) time delay jumper
51) controller

In circuit board figures Fig._22T:

A jumper shall control whether the time delay circuit receives power when the time control knob that activates power turns the power on. Thus a jumper can disable the time delay circuit.
Supply power to the switch common from the board power supply
Determine whether computer has power based on input from power cable.
If power is being supplied to power control indicators A, or B, or C, do not allow the board to switch anything, even if power to a power control indicator is changed, unless power is off to power input. For example, if power is being supplied to power indicator A, and then is switched to power indicator B while power is still being supplied to POWER INPUT, ignore the change and don't switch power to B. Only if power is removed from power input for 3 seconds or more, then switch to A, or B, or C.
The socket shall bypass the circuit board with the neutrals (see diagram of neutral jumper bypass).

A = Normal Mode
B = Repair Mode
C = Zap mode

FIG._18I

If switch is at A, don't change anything.
If switch is at B, turn on
if someone hit on button while zap is happening we are shot.

If Circuit A: Wait X continuous seconds and check to see if power is still on to A. If so, turn on time delay circuit. Leave circuit on Y seconds and then turn off circuit. If power is still on to B, don't run again until such time as power is removed from all controls A and B and C for at least 3 contiguous seconds.
If circuit C, turn on time delay after X seconds (follow b above) and then turn on 1,2,3,4.
ID Jumpers 4 and 6 are optional spares.
For multiple users/operating systems, and/or data storage devices, duplicate the circuitry in the drawing (except for the controller and switch/switch lock... in most cases only one controller and switch/switch lock is needed.

Fig._22U: One type of circuit board for Multi User System

Fig._22V One type of Circuit Board for Repair and Backup.

Optional Automatic Repair Example Script/program: On computer startup the script/program hides all activity that occurs in the background. During this time a logo and text is shown to user. This text and logo should be able to be modified in preferences. This "hiding" function can be toggled off/on at any time by keyboard input of a specific keys sequence: e.g. while the command key is down, sequential input of the letters: zappy

A program or script runs the following sequence of events (all in the background hidden from the user):

A backup program is executed that makes a complete backup of the data on the drive at ID 1. The destination of this backup should be able to be modified in user preferences. For example in could go in a partition or folder on the drive at ID 0, or could go on a drive at ID 2.

A program or script executes that checks to see that the backup has been made successfully. After confirmation that backup is complete and successful, user is given an optional dialog (that can be modified by client) allowing user to select one of the following options: no format of ID 1, quick format of ID1, low level format of ID1. (Background remains hidden) This dialog only shows up if selected in preferences. In lieu of this dialog preference can be selected in preferences as to which type of format, or lack thereof, is performed.
If user is give the option in preferences, based on user selection, quick, or low level, or no format is run on ID1. Otherwise whatever option is in preferences is done. Optional script then executes a program that copies some (not all) data on ID 0 to wherever they belong on ID 1. For example this script may copy such items as: Explorer Favorites, Nets cape Bookmarks, E-mail, in box, out box, and address book. Optional: LED and/or computer screen can provide a dialog box that says something like: please turn switch to "Normal Use" position. Optional: LED and/or computer screen can provide a dialog box that says something like: "please restart computer". Or those events can happen automatically. Optionally, all the events described above can also be written into ROM or the operating system.

Fig._22W One type of Circuit Board for Repair of Multi-User System Option.

FIG._18J

An optional time delay circuit can be integrated that has a circuit that is normally open. It can have two user controllable time delays controlled by knobs ranging from 5 seconds or less - 60 seconds or more. One time control knob shall control the amount of time until the circuit is closed. The second timer control knob shall control the length of time the circuit is open. A jumper shall control whether the time delay circuit receives power when the time control knob that activates power turns the power on. Thus a jumper can disable the time delay circuit.

If power is being supplied to power control indicators 1, or 2, or 3, or 4, do not allow the board to switch power, even if power to a power control indicator is changed. For example, if power is being supplied to power indicator 2, and then is switched to power indicator 4 while power is still being supplied to indicator 2, ignore the change and keep the terminals closed that are associated with power control indicator 2. Only if power is removed from all power control indicators for a period of 3 contiguous seconds or more, allow the changes in which circuits are closed when power is restored to one of the power control terminals.

Boards and switches combined must be any of these sizes or smaller (smaller is better). The socket shall bypass the circuit board with the neutrals (see diagram of neutral jumper bypass).

Power Control Indicator 1: When power is supplied to power control indicator #1: provide power to time delay circuit if jumper A is on. After time delay circuit has finished and power off for that circuit, provide power to input terminals 1, 2, 3, 4, 5, 6, 7, 8, 10. (not 9)
Power Control Indicator 2: When power is supplied to power control indicator #2: provide power to input terminals 3, 4, 6, 7, 8, 9, 10 (not 5)
Power Control Indicator 3: When power is supplied to power control indicator #3: provide power to time delay circuit if jumper A is on. After time delay circuit has finished and power off for that circuit, provide power to input terminals 11, 12, 13, 14, 15, 16, 17, 18, 20 (not 19)
Power Control Indicator 4: When power is supplied to power control indicator #4: provide power to time delay circuit if jumper A is on. After time delay circuit has finished and power off for that circuit, provide power to input terminals 13, 14, 16, 17, 18, 19, 20 (not 15)

Single User only uses power control indicators #1 and #2.
Please note: boards contain optional spare circuits that can be eliminated.
Please note: knobs for delay start and stop are temporary for the prototype and will be replaced with non-adjustable components after further experimentation with length of time for delay circuit. Only the material portions of the circuit board and socket are illustrated, and it will be understood that is not drawn to scale.

Fig._22Y: Example of Net-Lock system as external "add-on" device for network communications device (such as a network interface card, or modem, or other device) and a network. The data signal is redirected via the net-lock device to a switchlock (or other switch trigger) where the connection to the network can be locked "on" or "off" (wire circuit for data "opened" or "closed").

Fig._22Z: Example of cable connections being "redirected" through Switching System to enable switching each and/or all connections to either a state of "connected" or "disconnected" so as to change, for example, Device IDs, power, read/write abilities, data flow, lock/unlock status, etc. An optional logic controller is indicated to control the process as needed. Without a logic controller, switching could be controlled by, for example, a physical "toggle" switch.

Fig._22Z-1: Detailed example of cable connections being "redirected" through Switching Process to

FIG._18K

enable switching each and/or all connections to either a state of "connected" or "disconnected" so as to change, for example, Device IDs, power, read/write abilities, data flow, lock/unlock status, etc. An optional logic controller is indicated to control the process as needed. Without a logic controller, switching could be controlled by, for example, a physical "toggle" switch.

Fig._22Z-2: Example of Entertainment/Communication System utilizing modular blocks that "snap" into one another and thereby connect the wires that are embedded inside each block ("connectors"). Thus, by connecting blocks, circuits can be formed to create effects or functions (such as, for example, but not limited to, an LED and/or LCD and/or robotic device, connected via blocks, to a power supply).

Fig._22Z-3: Example of cable connections being "redirected" through Switching Process (in this example without using "wire connectors" as shown in Fig._22Z) to enable switching each and/or all connections to either a state of "connected" or "disconnected" so as to change, for example, Device IDs, power, read/write abilities, data flow, lock/unlock status, etc. An optional logic controller is indicated to control the process as needed. Without a logic controller, switching could be controlled by, for example, a physical "toggle" switch.

Fig._23A: Example of Switching Process switching jumper IDs so as to change the boot order of the attached data storage devices.

Fig._23B:

Fig._23C:

Fig._24 DIAGRAMS

Fig._24A: Example of a Switching System that utilizes a brake on the lock cylinder to prevent cylinder from moving when power is present.

Fig._24B: Example of Switching System that utilizes solenoid or piezo switch to stop lock cylinder from turning when power is present.

Fig._24C: Example of Switching System that utilizes either a brake, clutch, lock, piezo device, or solenoid, to prevent a rod from turning. The rod goes from a mechanical key or rotary switch to an electrical switch.

Fig._24D shows that the switching can be done on a circuit board represented by M2370, and switch by a switch and/or switchlock M2301. This Example of the Switching System that utilizes a programmable logic controller and/or circuit board to control one or more functions: for timing of switching, reset of PRAM, CUDA, CMOS, etc, control over switching, control over power, connection to network(s) for switch triggers and commands over network(s), control over switches and relays, control over circuit boards, control over global positioning system/security system, control over selection of startup device, control over switch triggering.

Fig._24E shows that the switching can be done by any sort of switching device and/or switch trigger.

We have used a simple switch in the drawings, but assume that anyone skilled in the arts can also diagram the switch into a circuit board if so desired, so that the switching function can stand

alone on a circuit board, or be integrated into other circuitry.
This is an example of the Switching System triggering methods.

Fig._26 DIAGRAMS

The diagram figures that begin with the letter "Fig._26" are examples to demonstrate that the switch(es) can be located anywhere, and on a wide range of devices, inline, and in wireless situations.
When computing equipment is used with any other type of device(s) (for example: robots, robotics, transports systems, televisions, telecommunications, manufacturing, equipment control, etc.) the switching system and/or switch trigger can be relocated, and/or additional switches and/or switch triggers added, so as to easily provide accessibility to users and/or people controlling the system. Additionally, the switch mechanism and/or trigger can be camouflaged or hidden as needed. Location, appearance, and type of switch can be changed as needed.

Fig._26A: Example of switch(es) located on computing hardware.

Fig._26B: Example of switch(es) located on computing hardware.

Fig._26C: Example of switch(es) located on robotic device.

Fig._26D: Example of switch(es) located on robotic device.

Fig._26E: Example of switch(es) located on robotic device.

Fig._26F: Example of switch(es) located on vehicle.

Fig._26G: Example of switch(es) located on vehicle.

Fig._26H: Example of switch(es) located on vehicle.

Fig._26I: Example of switch(es) located on vehicle.

Fig._26J: Example of switch(es) located on robotic device.

Fig._26K: Switching system integrated into data storage device. Please note that this is a functional design. The integrated Switching system can be integrated anywhere in or on the storage device, and/or integrated into the storage device circuitry. It can be wired and/or wireless, and an optional switch can be used that is separate and/or integrated into the data storage device itself.

Fig._26L: Switching System integrated into a StorExecute (see definition).

Fig._26M: Switching System integrated into a StorExecute (see definition).

Fig._26N: Switching System integrated into automatic CD burner that can create multiple burned CDs without user interaction.
Fig._26O: Front view of Switching System mounted in case.

Fig._26P: Front view of Switching System mounted in case.

Fig._26Q: Front view of Switching System mounted in case.

Fig._26R: Front view of Switching System mounted in case.

Fig._26S: Examples of LCD screen dialog.

Fig._26T: Examples of LCD screen dialog.

Fig._26U: Example of ability to disconnect a network connection with switch trigger mounted on vehicle dashboard.

Fig._25 DIAGRAMS

Fig._25A: Front View: Optional clear case sitting on top of computing device for housing components described herein.

Fig._25B: Side view: Optional clear case sitting on top of computing device for housing components described herein.

Fig._25C: Two versions of front View of acrylic block lit by LED or other lighting source.

Fig._25D, Fig._25E, Fig._25F, Fig._25G: Optional top and/or sides and/or back of computer/computing device/peripheral device with optional lip. The dots represent electrical connectors for providing electrical current.

Fig._25H: Clear case surrounding computer/computing device/peripheral device.

Fig._25I: Clear case surrounding computer/computing device/peripheral device. Optional area for liquids and pipes, wires, etc.

Fig._25J: Optional top and/or sides and/or back of computer/computing device/peripheral device. The dots and lines represent electrical connectors for providing electrical current.

Fig._25K and Fig._25L: Plugs into Fig._25J on one side, and is connected to Fig._25L on the other side. Main circuits are shown on Fig._25K, and can be routed by software to pin outs on Fig._25L.

Fig._25N: Optional acrylic (or similar plastic) lit by LED lights to indicate the state of the switch. Optionally it can also be used as a decoration triggered by sound, motion, etc. Various colored lights are used to change colors in acrylic. Other art pieces can also be put in these "entertainment" boxes, such as lava lights, plasma lamps, and various art projects.

Fig._25O: Optional top and/or side panels of computing device case with positive and negative connections.

Fig._25P: Optional Colored or clear Transparent Acrylic Case Lit by LED or other light source. A layer of white acrylic or similar material can be under the outer layer for the purpose of diffusing the light. Optionally, case can be clear and change color when computer functions change when different groups of colored LEDs go on depending on switching functions taking place, or sounds, motion or other triggers. LEDs can be hidden in bottom of case, which can be surrounded by colored material.

Backup & Repair	Freeze-Buster	Hardware Repair	Virus-Proof Hacker-Proof	Net-Lock	Multi-User	Entertainment/ Communication
25M	21Z-2	26A	21Z-7	21Z	21A	25A
25Q	26A	26B	21Z-8	21Z-1	21P	25J
21B	26B	L72	21Z-8	26A	21Z-3	25K
21C	26E	26F	21Z-9	26B	21Z-5	25L
21D	26G	26J	21Z-9	26D	26A	25M
21E	26J	24A	21Z-10	26I	26B	25B
21F	26P	24B	21P	26J	26C	25C
21G	26U	24C	21Q	26R	26H	25D
21H	24D	24D	21R	26U	26J	25E
21I	24E	24E	21T	24D	26K	25F
21J	20L	20L	21U	24E	26N	25G
21K	20R	20P	21V	20L	26Q	25H
21L	20T	20T	21W	20Q	26T	25N
21M	22M	20I	21X	20T	24A	25O
21N	22O	22U	21Y	22Y	24B	25P
21O		22I	21Z-3	W64.5	24C	25Q
21P		22J	21Z-6	22N	24D	25I
21Z-3		22K	26N		24E	20T
21Z-4		22L	24A		20J	22Z-2
26L		22Z-3	24B		20L	
26M			24C		20O	
26N			24D		20T	
26O			24E		20C	
26S			20J		20D	
24A			20K		20E	
24B			20L		20F	
24C			20M		20G	
24D			20N		22U	
24E			20O		22W	
20A			20S		22B	
20J			20T		22D	
20L			22P		22E	
20O			22R		22F	
20B			22S		22G	
20T			22Z		22H	
20H			22Z-1		22Q	
22T			22Z-3		22S	
22V	22Z				22Z	
22X	22Z-1				22Z-1	
22A	22Z-3				22Z-3	
22C						
22H						
23A						
WJ10						
23B						
23C						

TABLE OF WHICH DIAGRAMS GO WITH WHICH INVENTION EMBODIMENTS

FIG._19

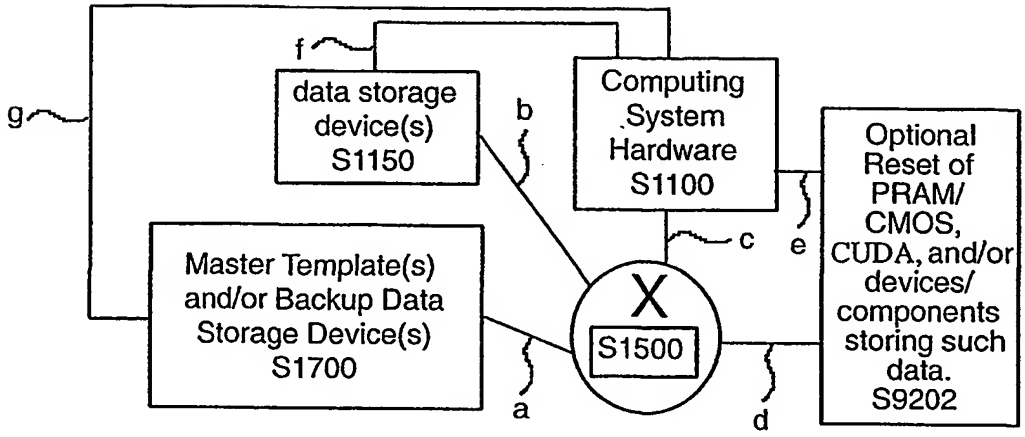


FIG._20A

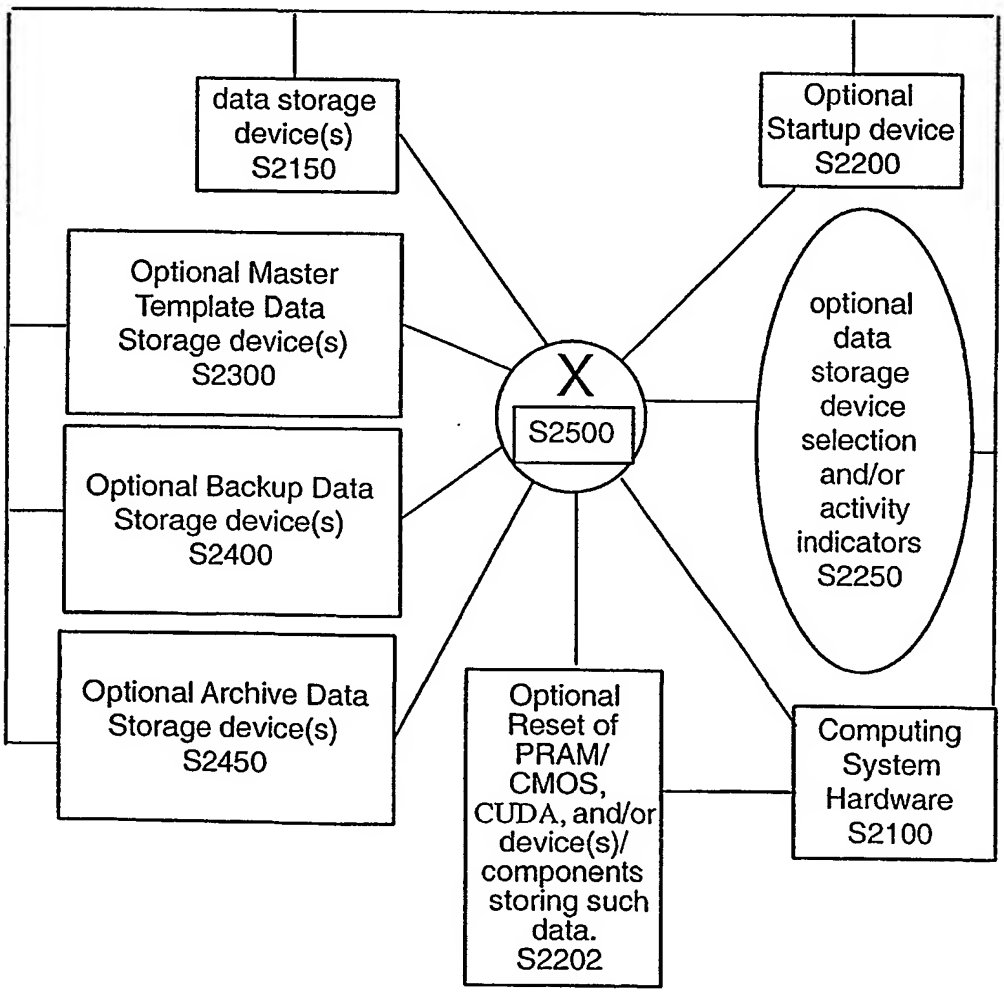


FIG._20B

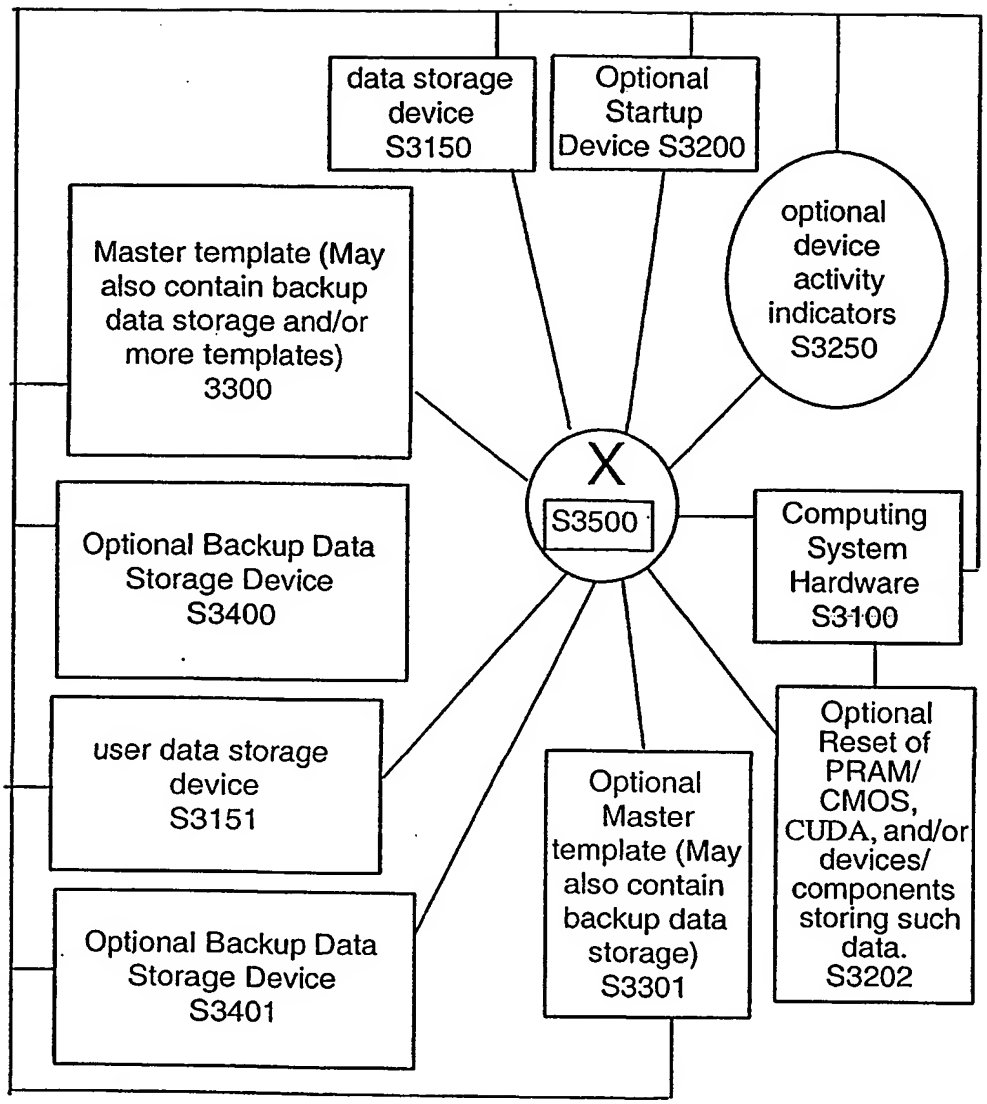

data storage device S3150
Optional Startup Device S3200
optional device activity indicators S3250
Master template (May also contain backup data storage and/or more templates) 3300
X
S3500
Computing System Hardware S3100
Optional Backup Data Storage Device S3400
user data storage device S3151
Optional Master template (May also contain backup data storage) S3301
Optional Reset of PRAM/ CMOS, CUDA, and/or devices/ components storing such data. S3202
Optional Backup Data Storage Device S3401

FIG._20C

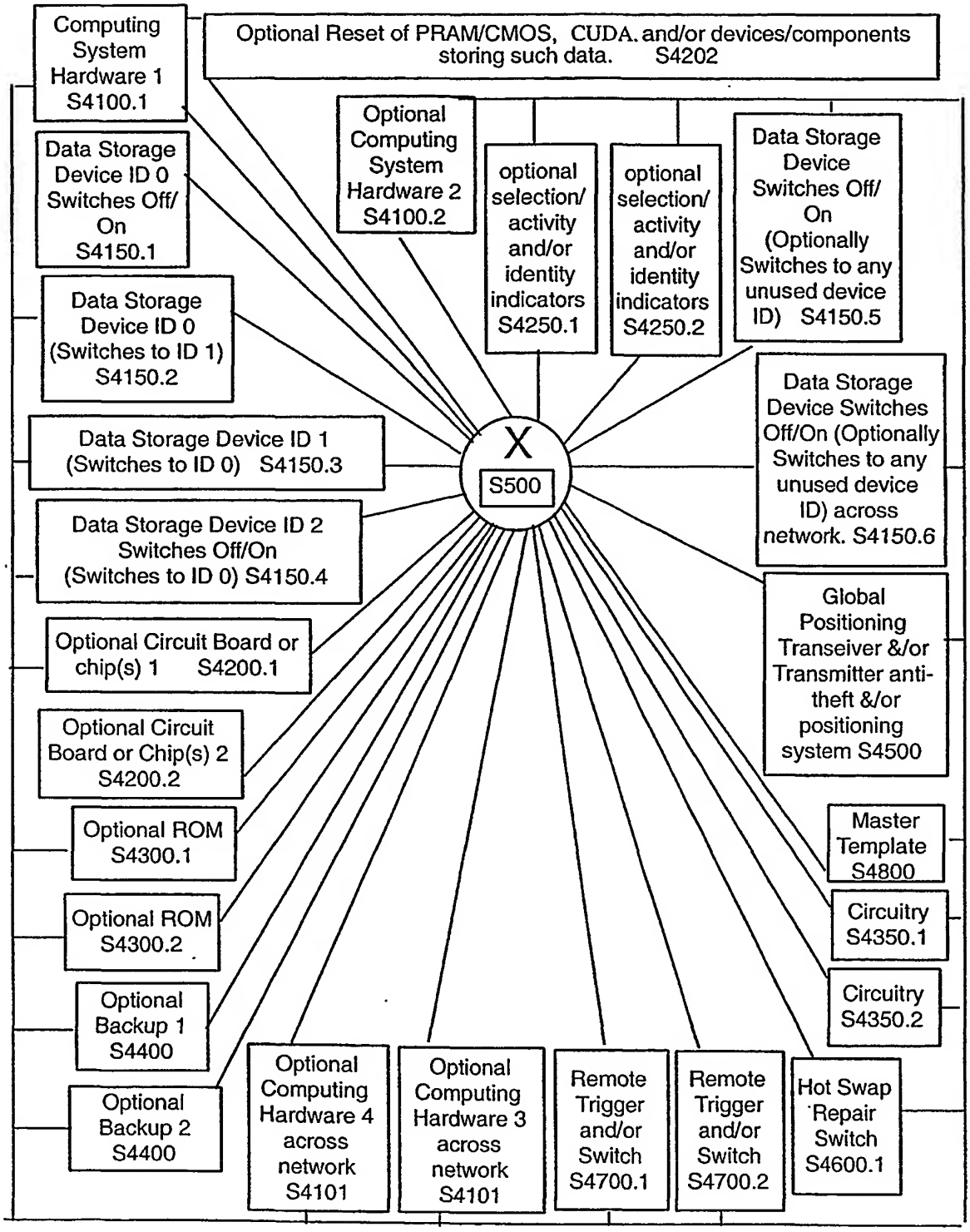

Computing System Hardware 1 S4100.1
Optional Reset of PRAM/CMOS, CUDA. and/or devices/components storing such data. S4202
Optional Computing System Hardware 2 S4100.2
optional selection/ activity and/or identity indicators S4250.1
optional selection/ activity and/or identity indicators S4250.2
Data Storage Device Switches Off/ On (Optionally Switches to any unused device ID) S4150.5
Data Storage Device ID 0 Switches Off/ On S4150.1
Data Storage Device ID 0 (Switches to ID 1) S4150.2
Data Storage Device ID 1 (Switches to ID 0) S4150.3
X
S500
Data Storage Device Switches Off/On (Optionally Switches to any unused device ID) across network. S4150.6
Data Storage Device ID 2 Switches Off/On (Switches to ID 0) S4150.4
Global Positioning Transeiver &/or Transmitter anti-theft &/or positioning system S4500
Optional Circuit Board or chip(s) 1 S4200.1
Optional Circuit Board or Chip(s) 2 S4200.2
Optional ROM S4300.1
Master Template S4800
Optional ROM S4300.2
Circuitry S4350.1
Optional Backup 1 S4400
Circuitry S4350.2
Optional Backup 2 S4400
Optional Computing Hardware 4 across network S4101
Optional Computing Hardware 3 across network S4101
Remote Trigger and/or Switch S4700.1
Remote Trigger and/or Switch S4700.2
Hot Swap Repair Switch S4600.1

FIG._20D

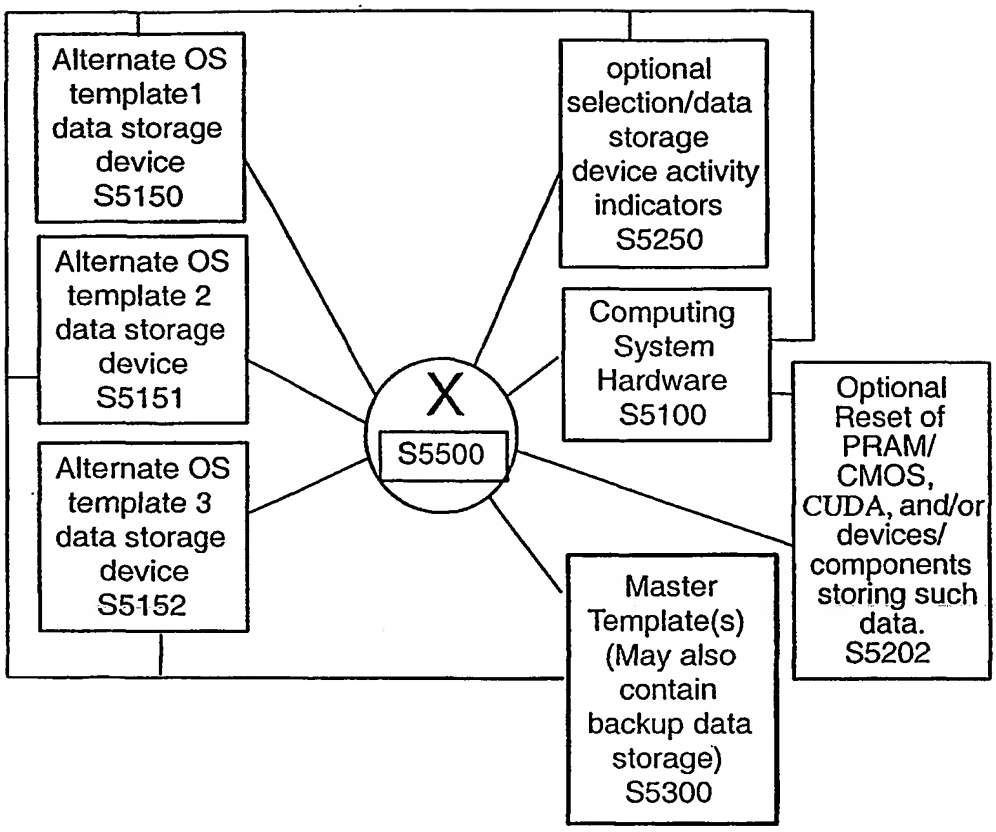

Alternate OS template1 data storage device S5150
Alternate OS template 2 data storage device S5151
Alternate OS template 3 data storage device S5152
X
S5500
optional selection/data storage device activity indicators S5250
Computing System Hardware S5100
Optional Reset of PRAM/ CMOS, CUDA, and/or devices/ components storing such data. S5202
Master Template(s) (May also contain backup data storage) S5300

FIG._20E

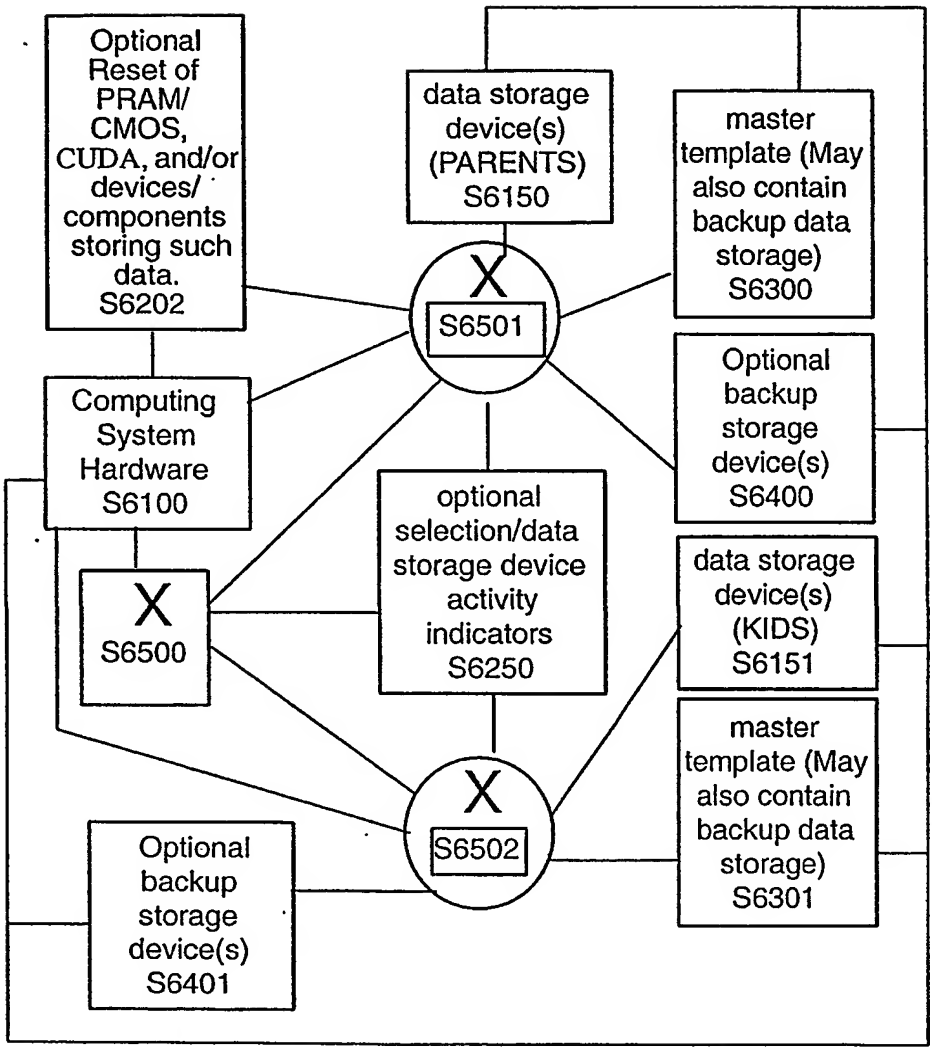

Optional Reset of PRAM/ CMOS, CUDA, and/or devices/ components storing such data. S6202
data storage device(s) (PARENTS) S6150
master template (May also contain backup data storage) S6300
X
S6501
Optional backup storage device(s) S6400
Computing System Hardware S6100
optional selection/data storage device activity indicators S6250
X
S6500
data storage device(s) (KIDS) S6151
master template (May also contain backup data storage) S6301
X
S6502
Optional backup storage device(s) S6401

FIG._20F

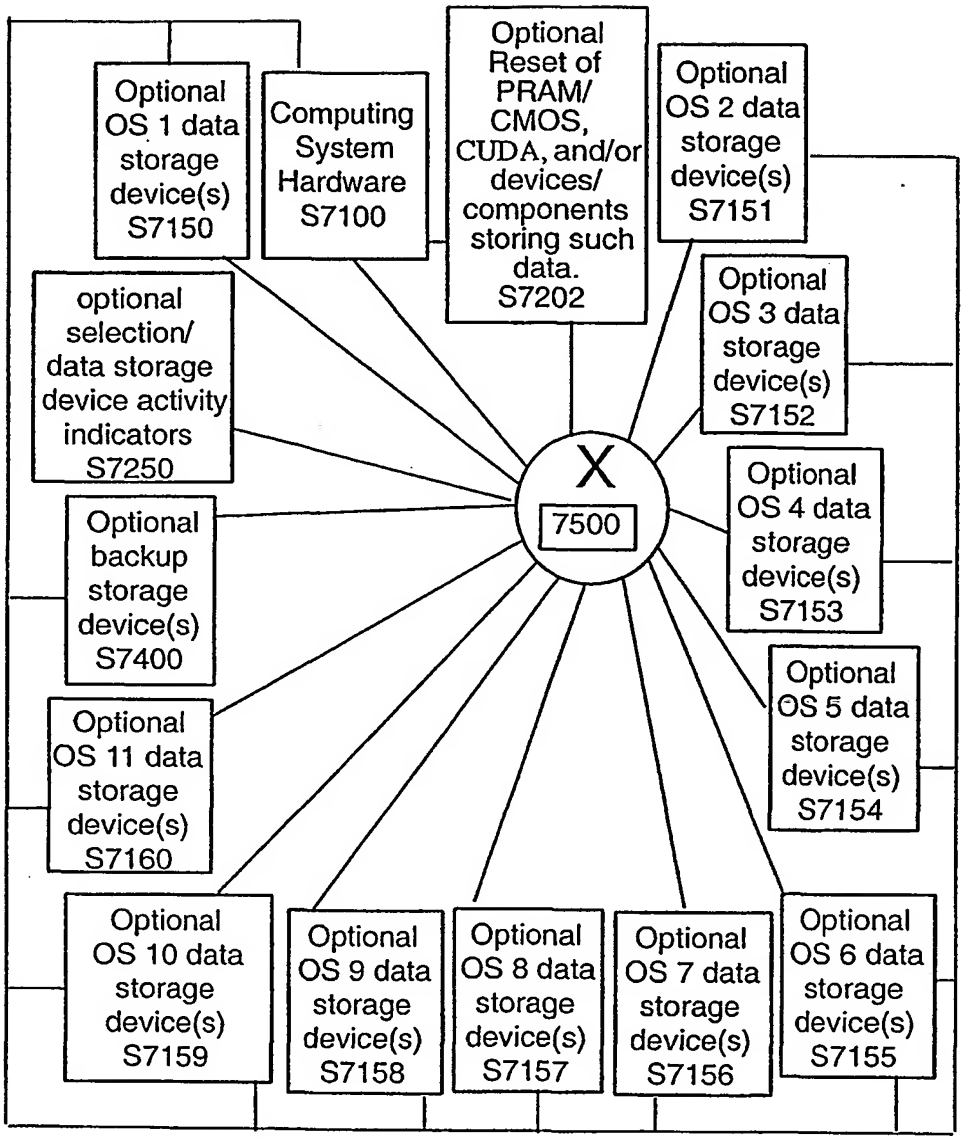

Optional OS 1 data storage device(s) S7150
Computing System Hardware S7100
Optional Reset of PRAM/ CMOS, CUDA, and/or devices/ components storing such data. S7202
Optional OS 2 data storage device(s) S7151
optional selection/ data storage device activity indicators S7250
Optional OS 3 data storage device(s) S7152
X
7500
Optional backup storage device(s) S7400
Optional OS 4 data storage device(s) S7153
Optional OS 5 data storage device(s) S7154
Optional OS 11 data storage device(s) S7160
Optional OS 10 data storage device(s) S7159
Optional OS 9 data storage device(s) S7158
Optional OS 8 data storage device(s) S7157
Optional OS 7 data storage device(s) S7156
Optional OS 6 data storage device(s) S7155

FIG._20G

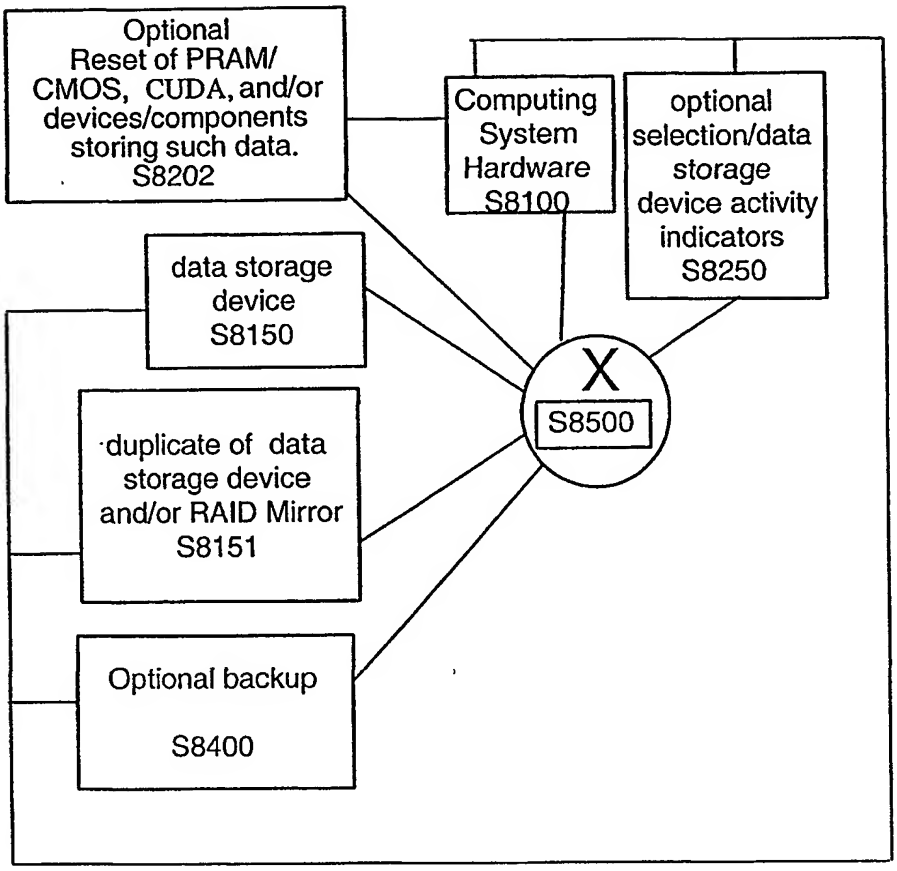

Optional
Reset of PRAM/
CMOS, CUDA, and/or
devices/components
storing such data.
S8202
Computing
System
Hardware
S8100
optional
selection/data
storage
device activity
indicators
S8250
data storage
device
S8150
X
S8500
duplicate of data
storage device
and/or RAID Mirror
S8151
Optional backup
S8400

FIG._20H

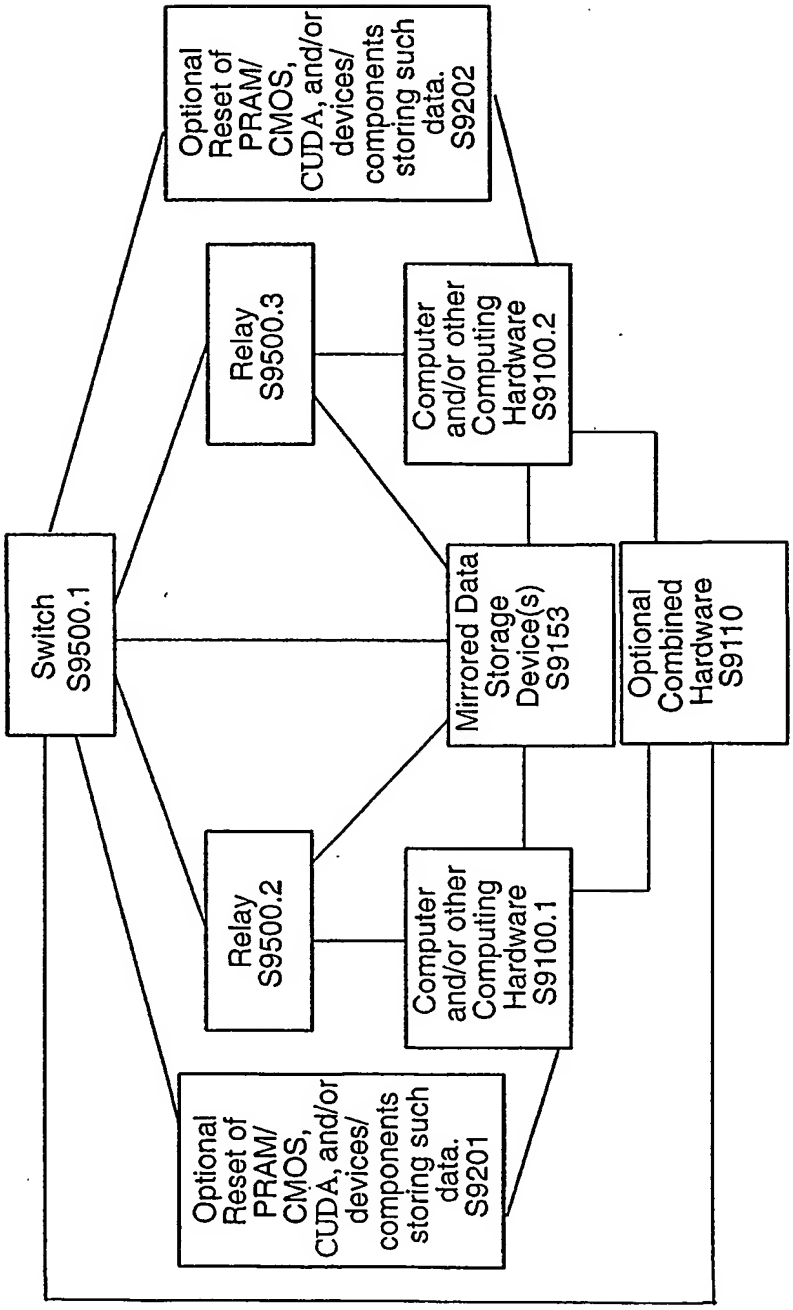

Optional Reset of PRAM/ CMOS, CUDA, and/or devices/ components storing such data. S9202
Relay S9500.3
Computer and/or other Computing Hardware S9100.2
Switch S9500.1
Mirrored Data Storage Device(s) S9153
Optional Combined Hardware S9110
Relay S9500.2
Computer and/or other Computing Hardware S9100.1
Optional Reset of PRAM/ CMOS, CUDA, and/or devices/ components storing such data. S9201

FIG._20I

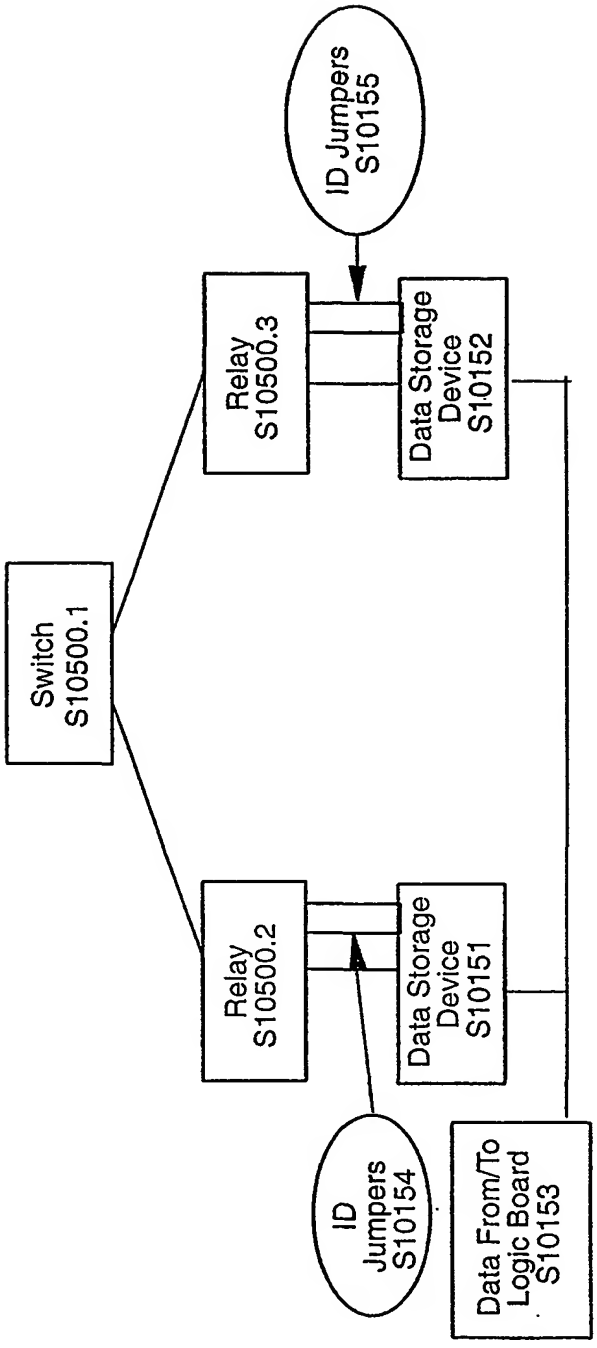

ID Jumpers S10155
Relay S10500.3
Data Storage Device S10152
Switch S10500.1
Relay S10500.2
Data Storage Device S10151
ID Jumpers S10154
Data From/To Logic Board S10153

FIG._20J

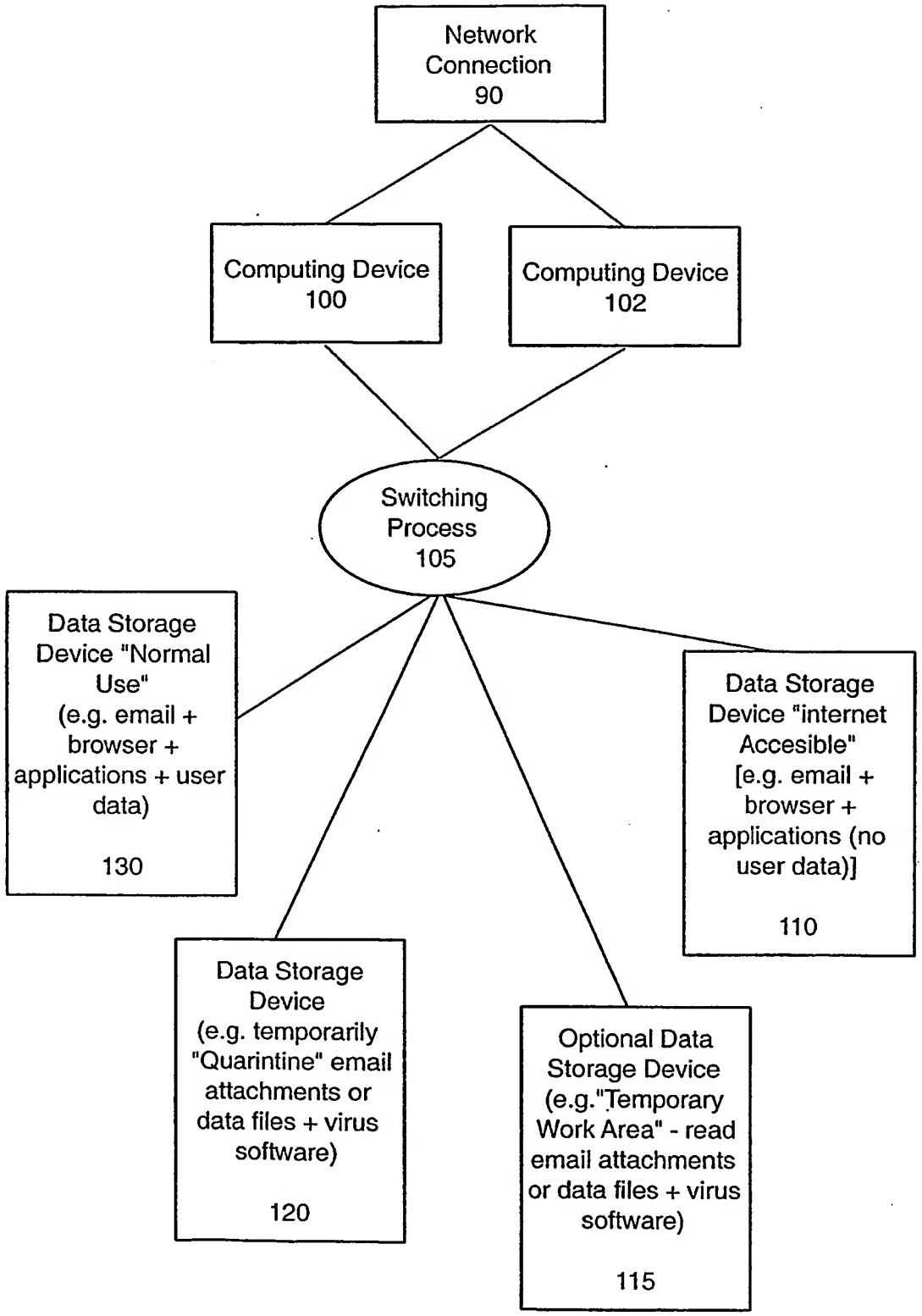

Network Connection 90
Computing Device 100
Computing Device 102
Switching Process 105
Data Storage Device "Normal Use" (e.g. email + browser + applications + user data) 130
Data Storage Device "internet Accesible" [e.g. email + browser + applications (no user data)] 110
Data Storage Device (e.g. temporarily "Quarintine" email attachments or data files + virus software) 120
Optional Data Storage Device (e.g."Temporary Work Area" - read email attachments or data files + virus software) 115

FIG._20K

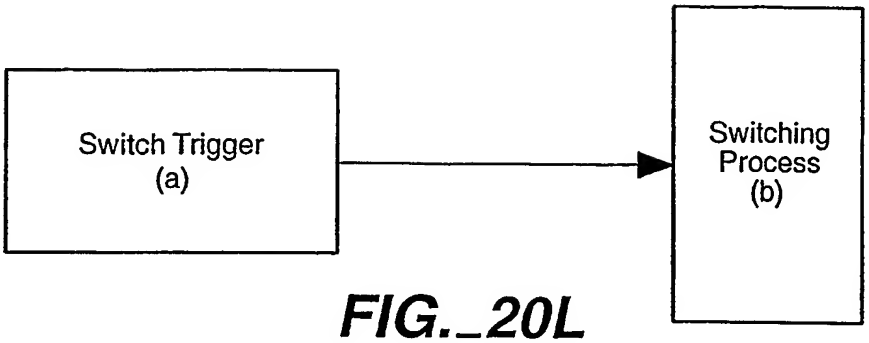


FIG._20L

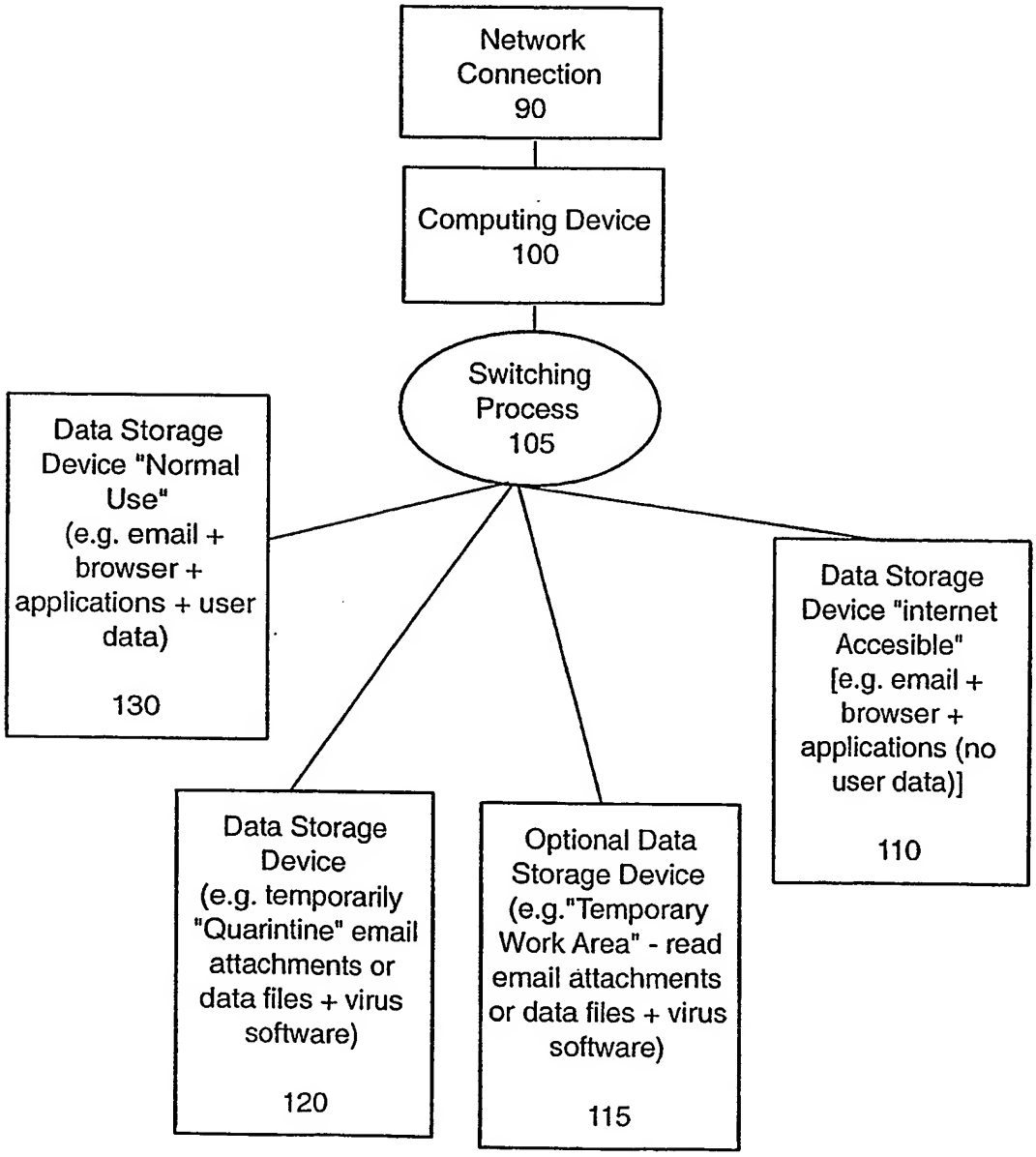
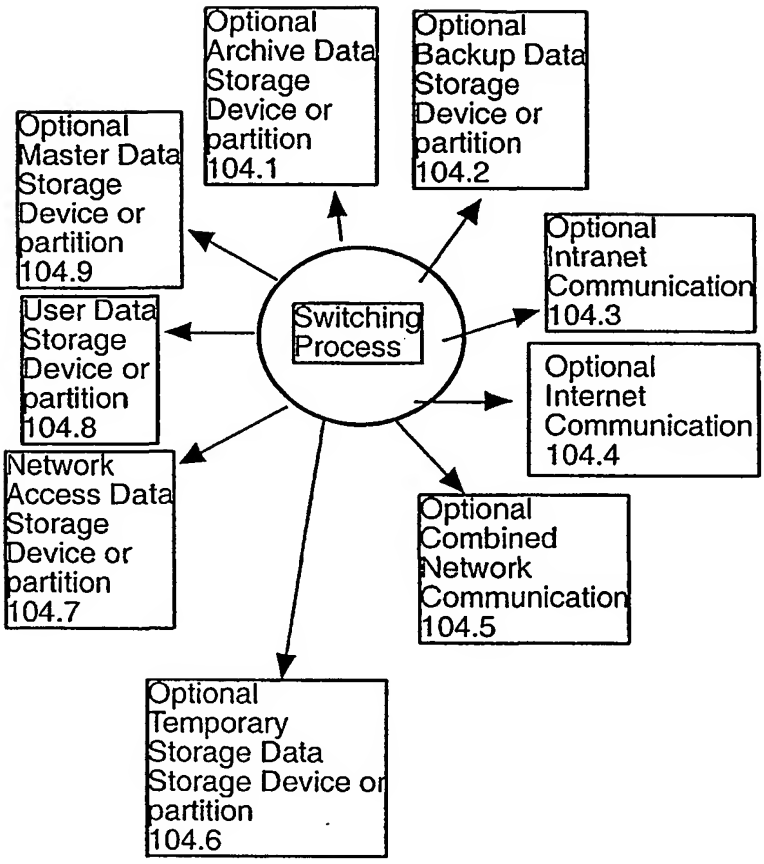


FIG._20M



Hacking and Virus Proof Computing Device 104.10

FIG._20N

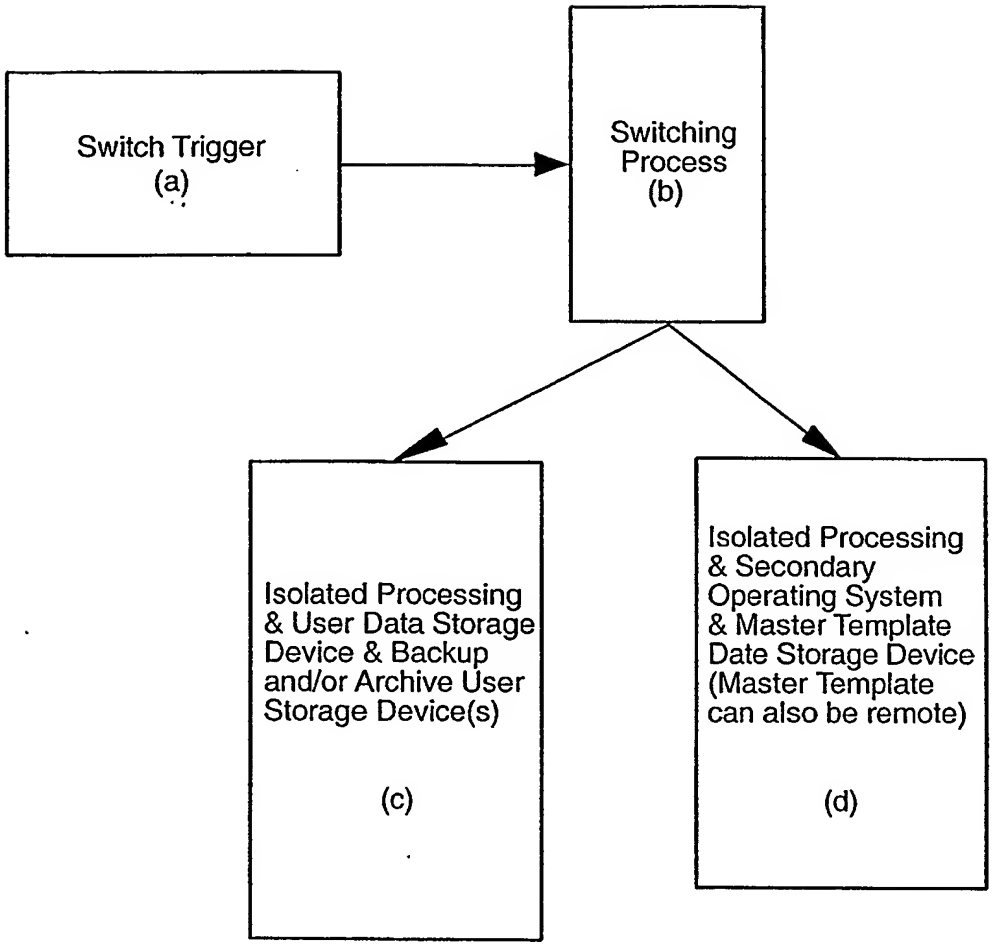

Switch Trigger
(a)
Switching
Process
(b)
Isolated Processing
& User Data Storage
Device & Backup
and/or Archive User
Storage Device(s)
(c)
Isolated Processing
& Secondary
Operating System
& Master Template
Date Storage Device
(Master Template
can also be remote)
(d)

FIG._20O

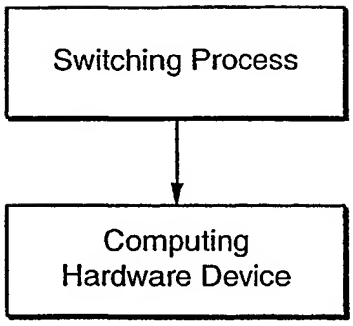


FIG._20P

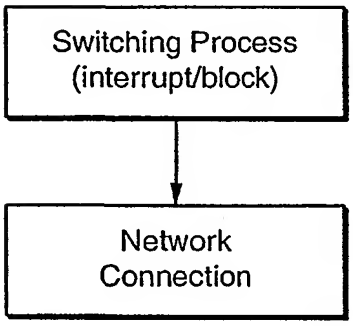


FIG._20Q

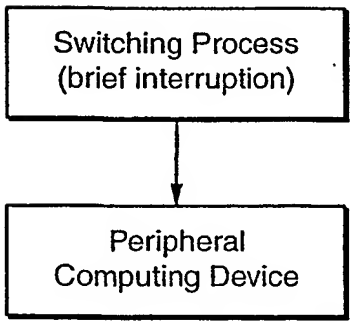


FIG._20R

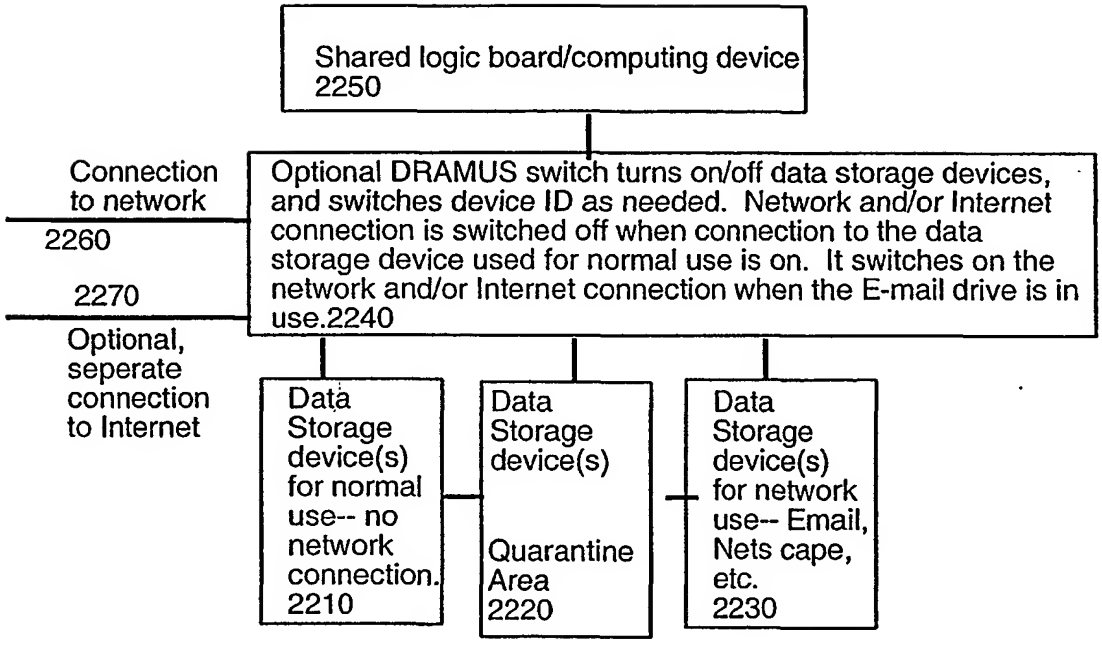

Shared logic board/computing device 2250
Connection to network
2260
2270
Optional, seperate connection to Internet
Optional DRAMUS switch turns on/off data storage devices, and switches device ID as needed. Network and/or Internet connection is switched off when connection to the data storage device used for normal use is on. It switches on the network and/or Internet connection when the E-mail drive is in use.2240
Data Storage device(s) for normal use-- no network connection. 2210
Data Storage device(s) Quarantine Area 2220
Data Storage device(s) for network use-- Email, Nets cape, etc. 2230

FIG._20S

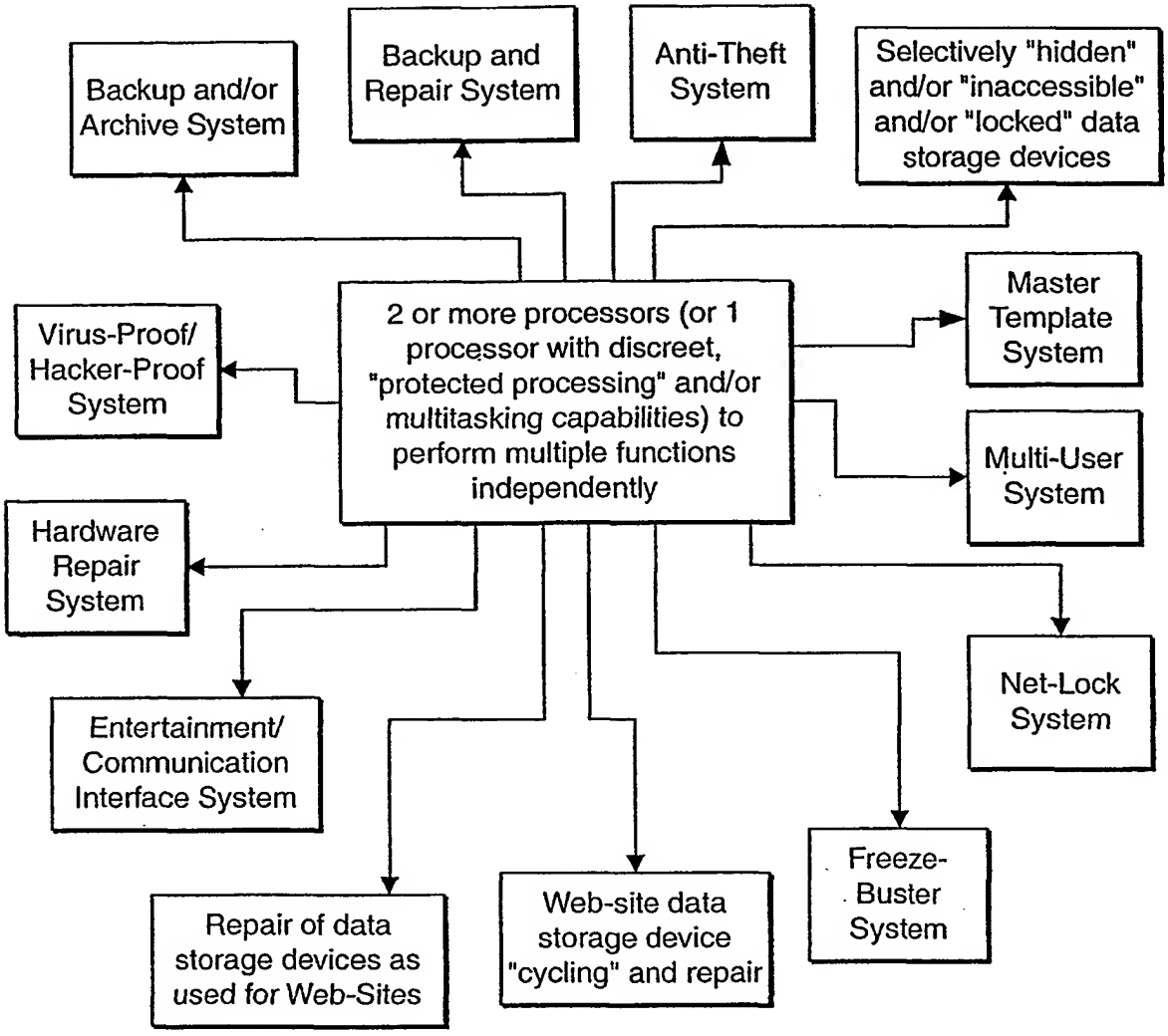

Backup and/or Archive System
Backup and Repair System
Anti-Theft System
Selectively "hidden" and/or "inaccessible" and/or "locked" data storage devices
Virus-Proof/ Hacker-Proof System
2 or more processors (or 1 processor with discreet, "protected processing" and/or multitasking capabilities) to perform multiple functions independently
Master Template System
Multi-User System
Hardware Repair System
Net-Lock System
Entertainment/ Communication Interface System
Freeze-Buster System
Repair of data storage devices as used for Web-Sites
Web-site data storage device "cycling" and repair

FIG._20T

Multi user

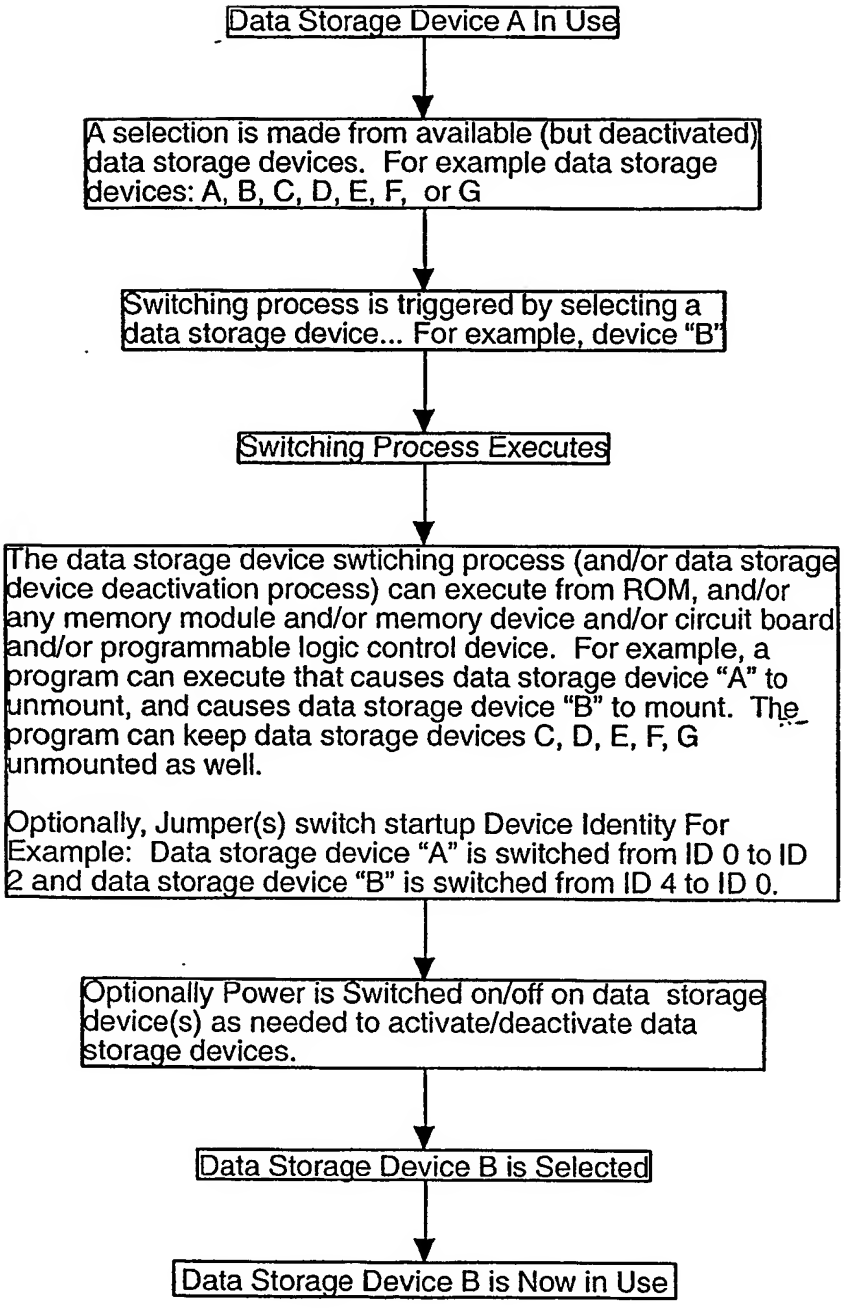


FIG._21A

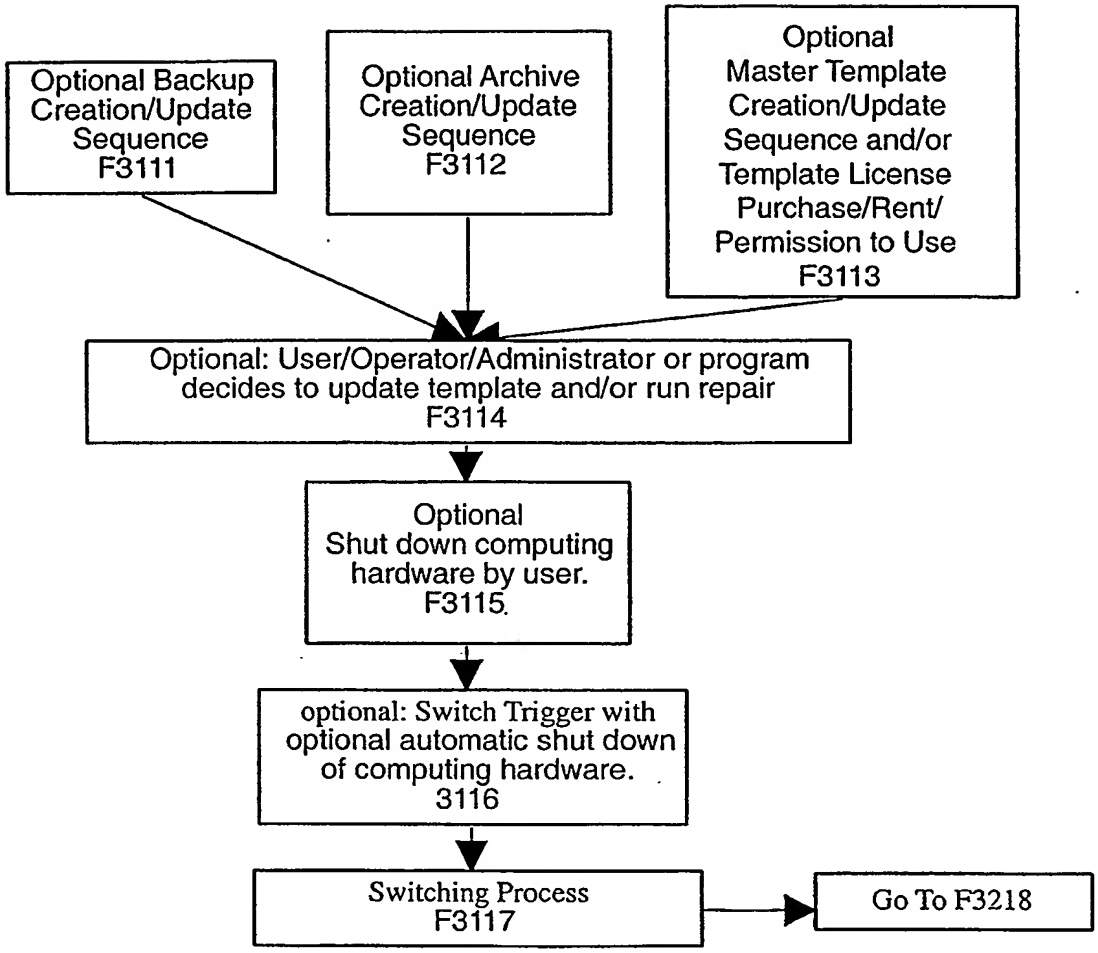

Optional Backup Creation/Update Sequence F3111
Optional Archive Creation/Update Sequence F3112
Optional Master Template Creation/Update Sequence and/or Template License Purchase/Rent/ Permission to Use F3113
Optional: User/Operator/Administrator or program decides to update template and/or run repair F3114
Optional Shut down computing hardware by user. F3115.
optional: Switch Trigger with optional automatic shut down of computing hardware. 3116
Switching Process F3117
Go To F3218

FIG._21B

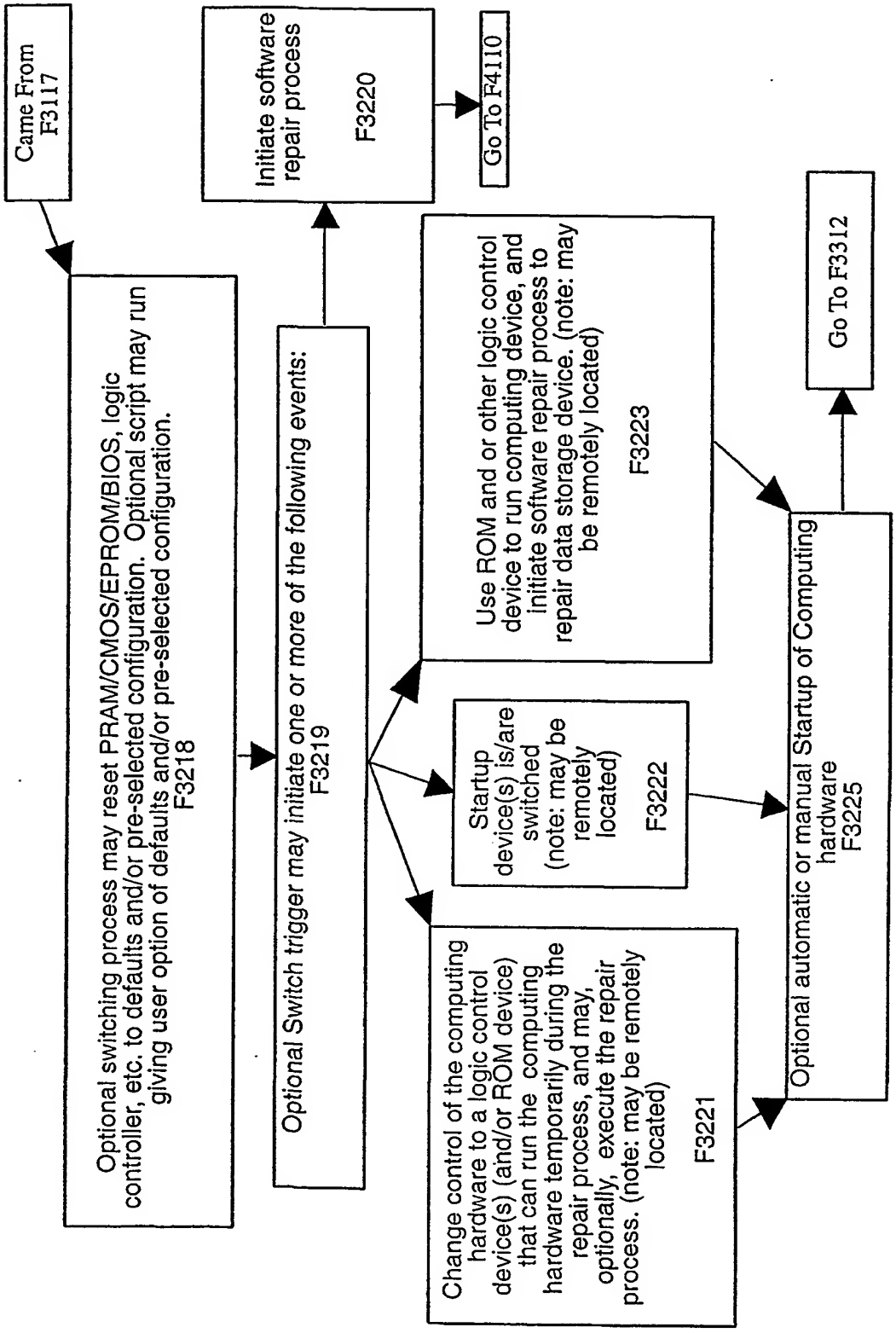

Came From F3117
Optional switching process may reset PRAM/CMOS/EPROM/BIOS, logic controller, etc. to defaults and/or pre-selected configuration. Optional script may run giving user option of defaults and/or pre-selected configuration. F3218
Optional Switch trigger may initiate one or more of the following events: F3219
Initiate software repair process F3220
Go To F4110
Use ROM and or other logic control device to run computing device, and initiate software repair process to repair data storage device. (note: may be remotely located) F3223
Startup device(s) is/are switched (note: may be remotely located) F3222
Change control of the computing hardware to a logic control device(s) (and/or ROM device) that can run the computing hardware temporarily during the repair process, and may, optionally, execute the repair process. (note: may be remotely located) F3221
Optional automatic or manual Startup of Computing hardware F3225
Go To F3312

FIG._21C

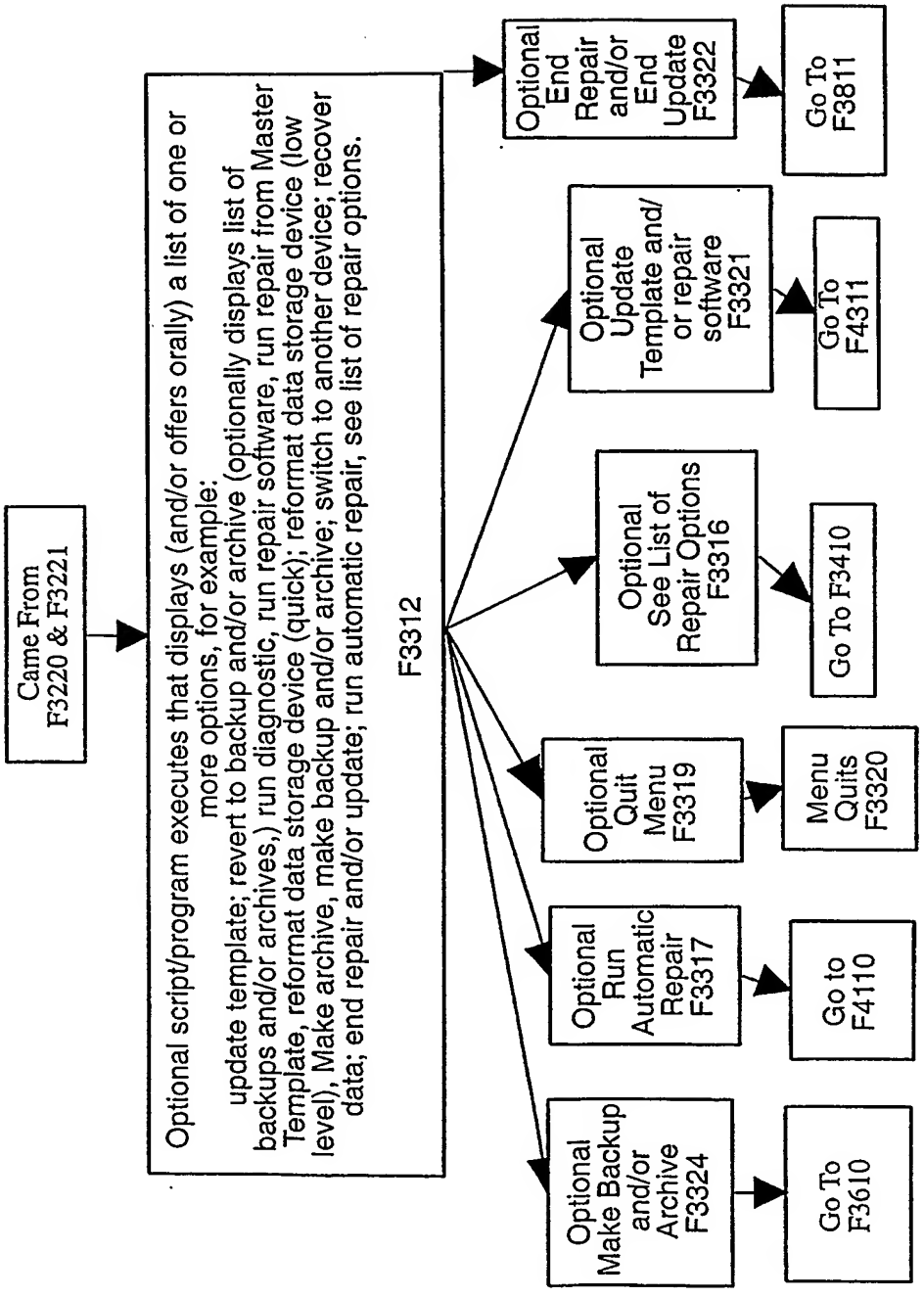

Came From F3220 & F3221
Optional script/program executes that displays (and/or offers orally) a list of one or more options, for example:
update template; revert to backup and/or archive (optionally displays list of backups and/or archives,) run diagnostic, run repair software, run repair from Master Template, reformat data storage device (quick); reformat data storage device (low level), Make archive, make backup and/or archive; switch to another device; recover data; end repair and/or update; run automatic repair, see list of repair options.
F3312
Optional Make Backup and/or Archive F3324
Go To F3610
Optional Run Automatic Repair F3317
Go to F4110
Optional Quit Menu F3319
Menu Quits F3320
Optional See List of Repair Options F3316
Go To F3410
Optional Update Template and/ or repair software F3321
Go To F4311
Optional End Repair and/or End Update F3322
Go To F3811

FIG._21D

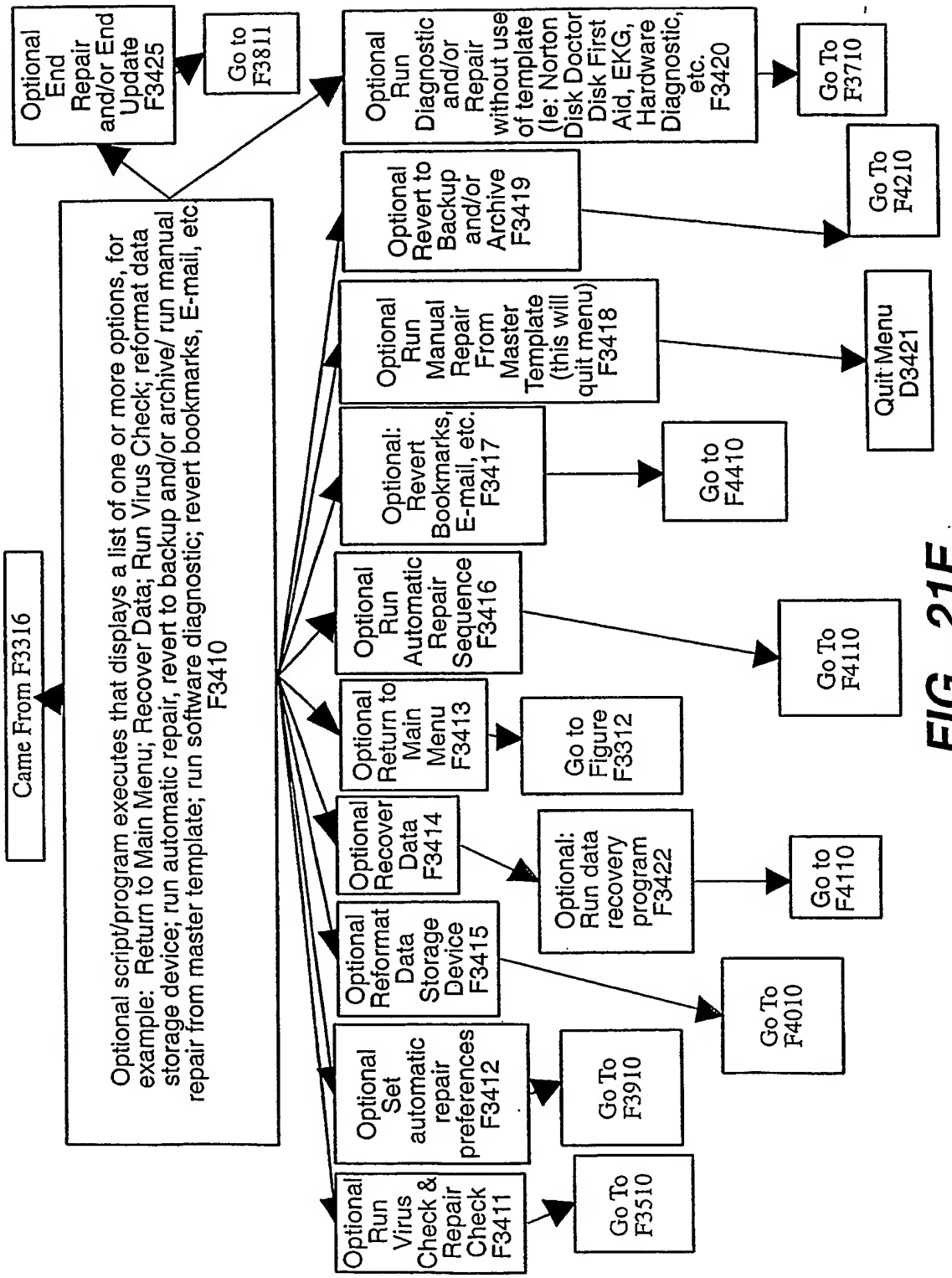

Came From F3316
Optional script/program executes that displays a list of one or more options, for example: Return to Main Menu; Recover Data; Run Virus Check; reformat data storage device; run automatic repair, revert to backup and/or archive/ run manual repair from master template; run software diagnostic; revert bookmarks, E-mail, etc F3410
Optional End Repair and/or End Update F3425
Go to F3811
Optional Run Virus Check & Repair Check F3411
Go To F3510
Optional Set automatic repair preferences F3412
Go To F3910
Optional Reformat Data Storage Device F3415
Go To F4010
Optional Recover Data F3414
Optional: Run data recovery program F3422
Go to F4110
Optional Return to Main Menu F3413
Go to Figure F3312
Optional Run Automatic Repair Sequence F3416
Go To F4110
Optional: Revert Bookmarks, E-mail, etc. F3417
Go to F4410
Optional Run Manual Repair From Master Template (this will quit menu) F3418
Quit Menu D3421
Optional Revert to Backup and/or Archive F3419
Go To F4210
Optional Run Diagnostic and/or Repair without use of template (Ie: Norton Disk Doctor Disk First Aid, EKG, Hardware Diagnostic, etc. F3420
Go To F3710

FIG._21E

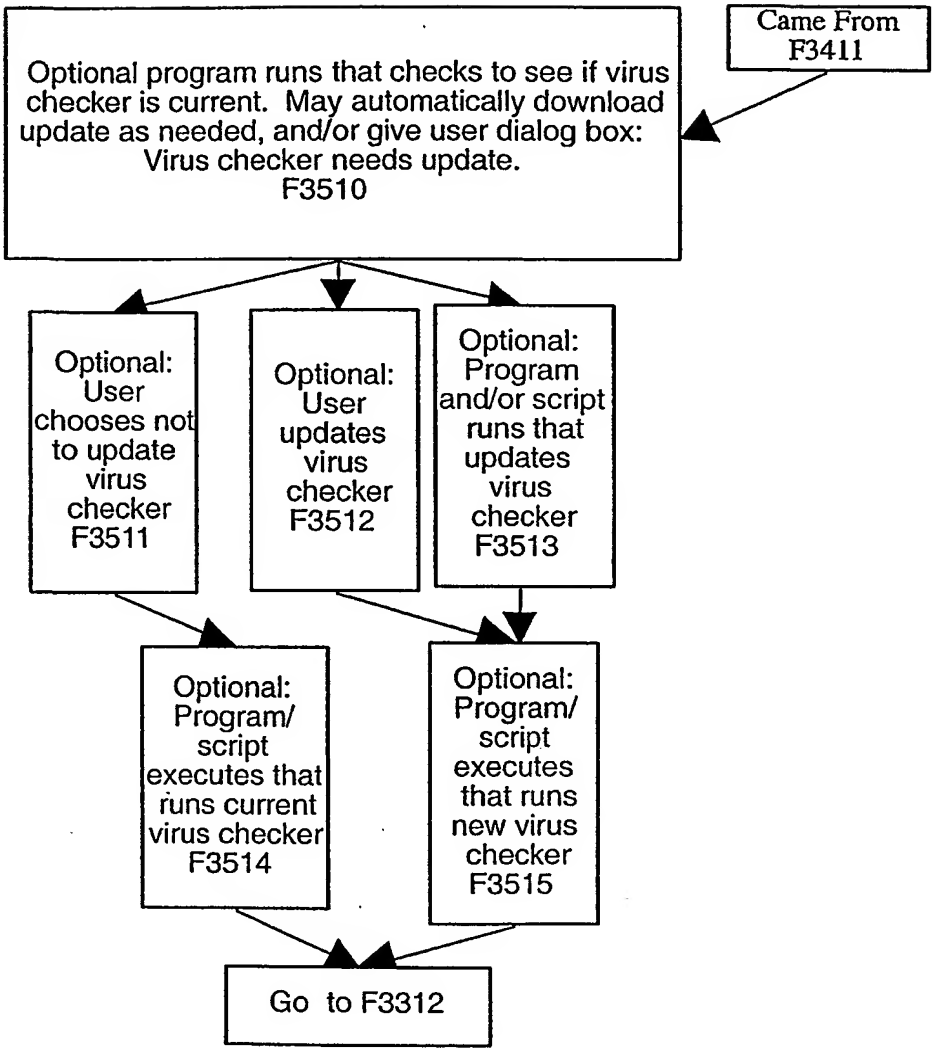

Came From F3411
Optional program runs that checks to see if virus checker is current. May automatically download update as needed, and/or give user dialog box: Virus checker needs update. F3510
Optional: User chooses not to update virus checker F3511
Optional: User updates virus checker F3512
Optional: Program and/or script runs that updates virus checker F3513
Optional: Program/ script executes that runs current virus checker F3514
Optional: Program/ script executes that runs new virus checker F3515
Go to F3312

FIG._21F

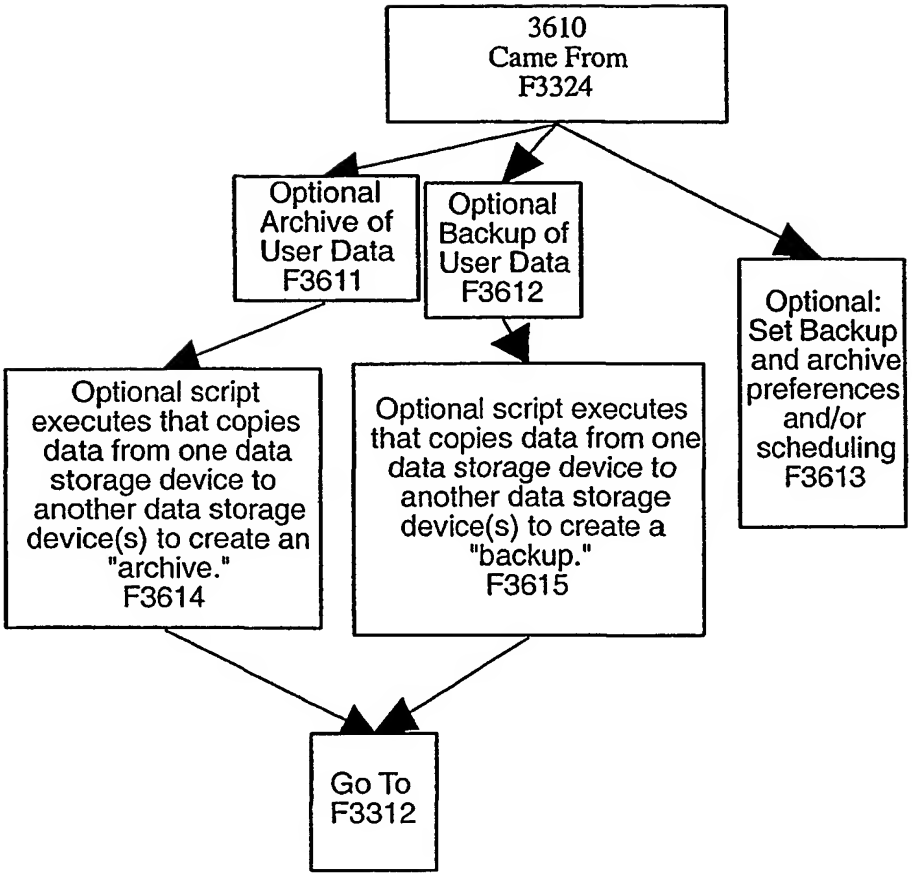

3610
Came From
F3324
Optional
Archive of
User Data
F3611
Optional
Backup of
User Data
F3612
Optional:
Set Backup
and archive
preferences
and/or
scheduling
F3613
Optional script
executes that copies
data from one data
storage device to
another data storage
device(s) to create an
"archive."
F3614
Optional script executes
that copies data from one
data storage device to
another data storage
device(s) to create a
"backup."
F3615
Go To
F3312

FIG._21G

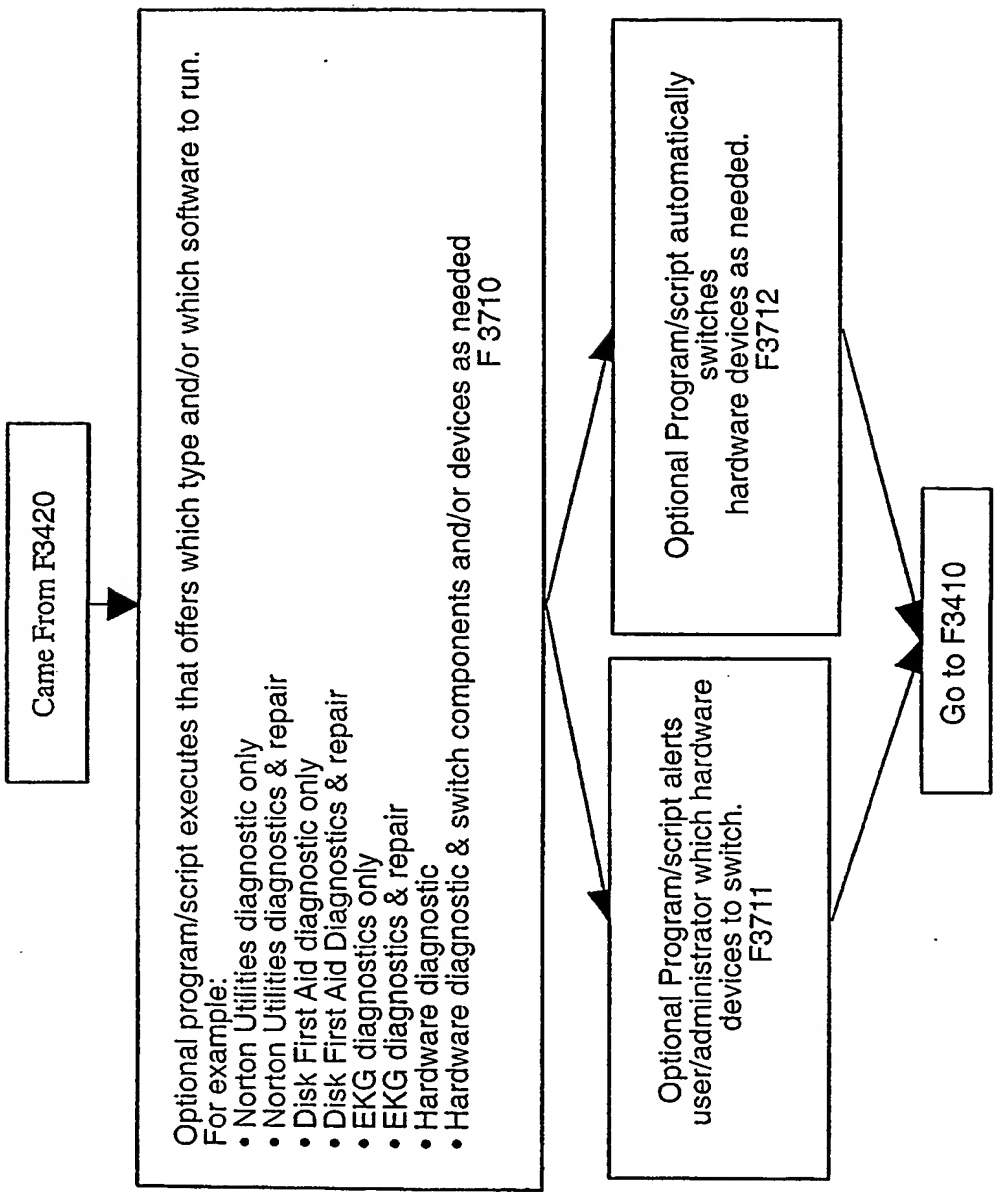

Came From F3420
Optional program/script executes that offers which type and/or which software to run.
For example:
• Norton Utilities diagnostic only
• Norton Utilities diagnostics & repair
• Disk First Aid diagnostic only
• Disk First Aid Diagnostics & repair
• EKG diagnostics only
• EKG diagnostics & repair
• Hardware diagnostic
• Hardware diagnostic & switch components and/or devices as needed
F 3710
Optional Program/script alerts user/administrator which hardware devices to switch.
F3711
Optional Program/script automatically switches
hardware devices as needed.
F3712
Go to F3410

FIG._21H

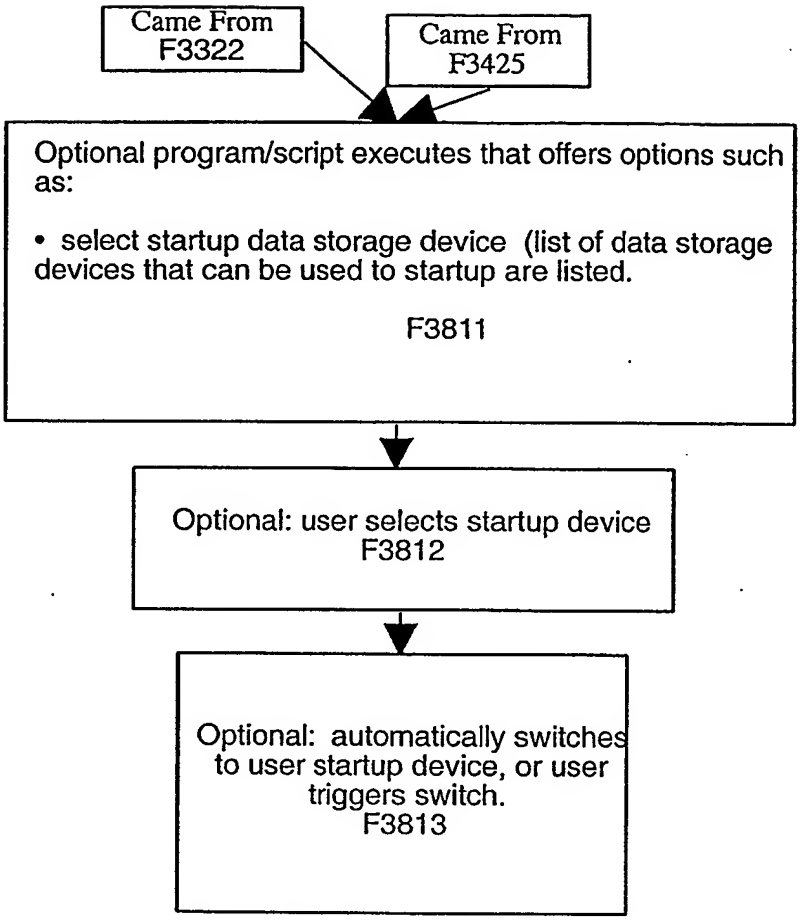

Came From F3322
Came From F3425
Optional program/script executes that offers options such as:
• select startup data storage device (list of data storage devices that can be used to startup are listed.
F3811
Optional: user selects startup device F3812
Optional: automatically switches to user startup device, or user triggers switch. F3813

FIG._21I

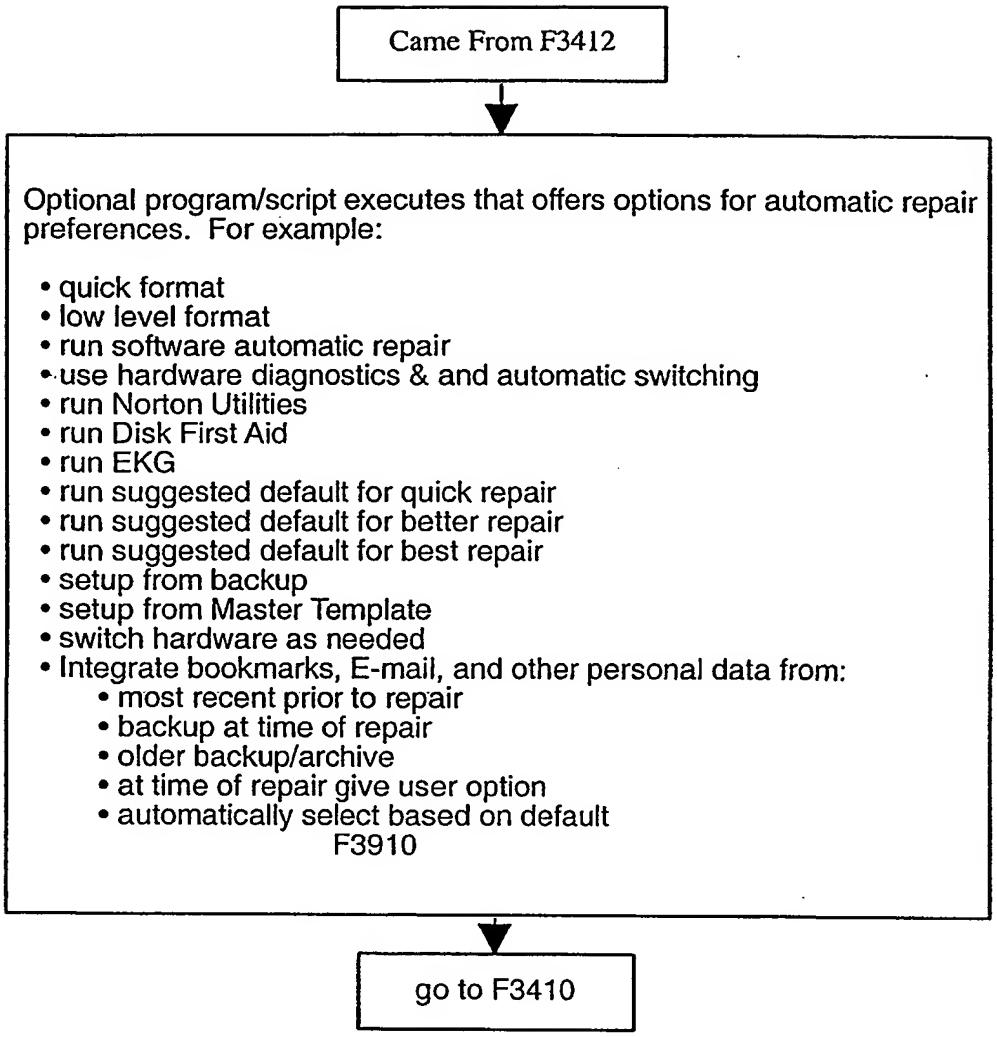

Came From F3412
Optional program/script executes that offers options for automatic repair preferences. For example:
• quick format
• low level format
• run software automatic repair
• use hardware diagnostics & and automatic switching
• run Norton Utilities
• run Disk First Aid
• run EKG
• run suggested default for quick repair
• run suggested default for better repair
• run suggested default for best repair
• setup from backup
• setup from Master Template
• switch hardware as needed
• Integrate bookmarks, E-mail, and other personal data from:
• most recent prior to repair
• backup at time of repair
• older backup/archive
• at time of repair give user option
• automatically select based on default
F3910
go to F3410

FIG._21J

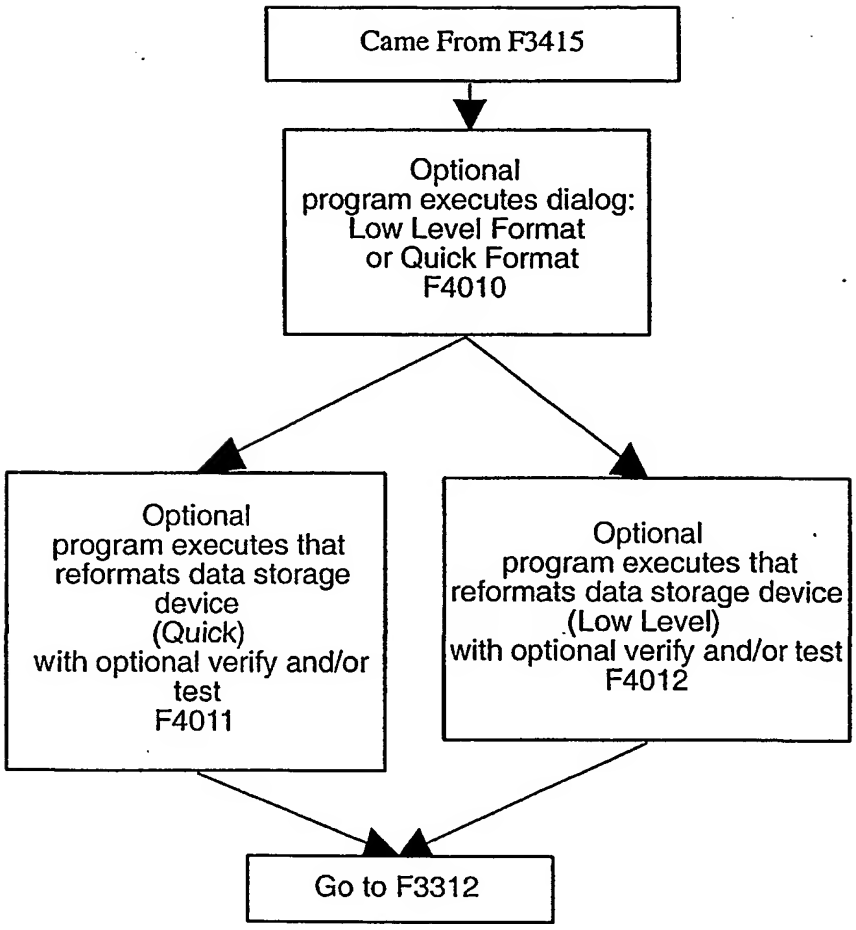

Came From F3415
Optional program executes dialog: Low Level Format or Quick Format F4010
Optional program executes that reformats data storage device (Quick) with optional verify and/or test F4011
Optional program executes that reformats data storage device (Low Level) with optional verify and/or test F4012
Go to F3312

FIG._21K

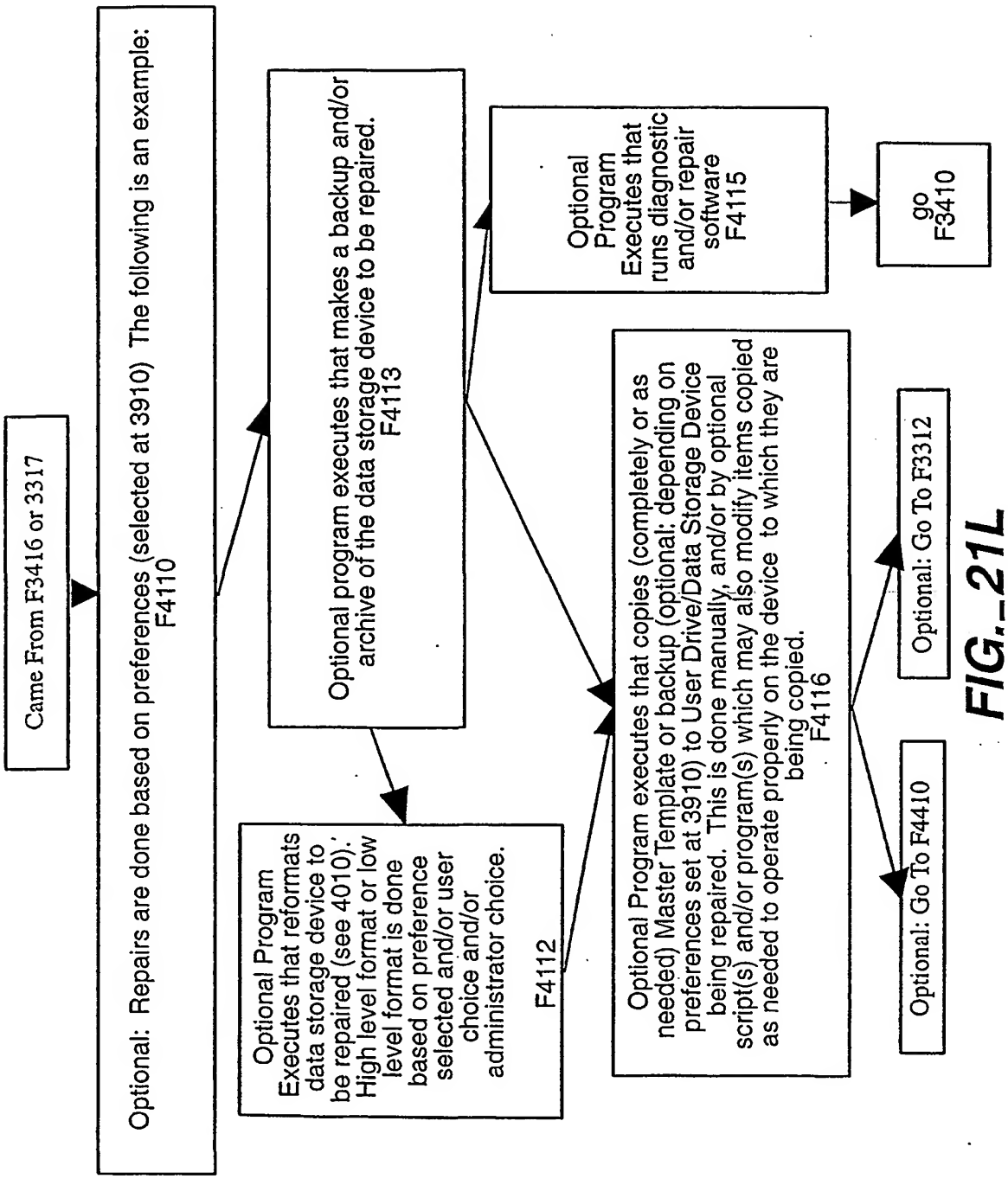

Came From F3416 or 3317
Optional: Repairs are done based on preferences (selected at 3910) The following is an example: F4110
Optional program executes that makes a backup and/or archive of the data storage device to be repaired. F4113
Optional Program Executes that reformats data storage device to be repaired (see 4010). High level format or low level format is done based on preference selected and/or user choice and/or administrator choice. F4112
Optional Program Executes that runs diagnostic and/or repair software F4115
go F3410
Optional Program executes that copies (completely or as needed) Master Template or backup (optional: depending on preferences set at 3910) to User Drive/Data Storage Device being repaired. This is done manually, and/or by optional script(s) and/or program(s) which may also modify items copied as needed to operate properly on the device to which they are being copied. F4116
Optional: Go To F3312
Optional: Go To F4410

FIG._21L

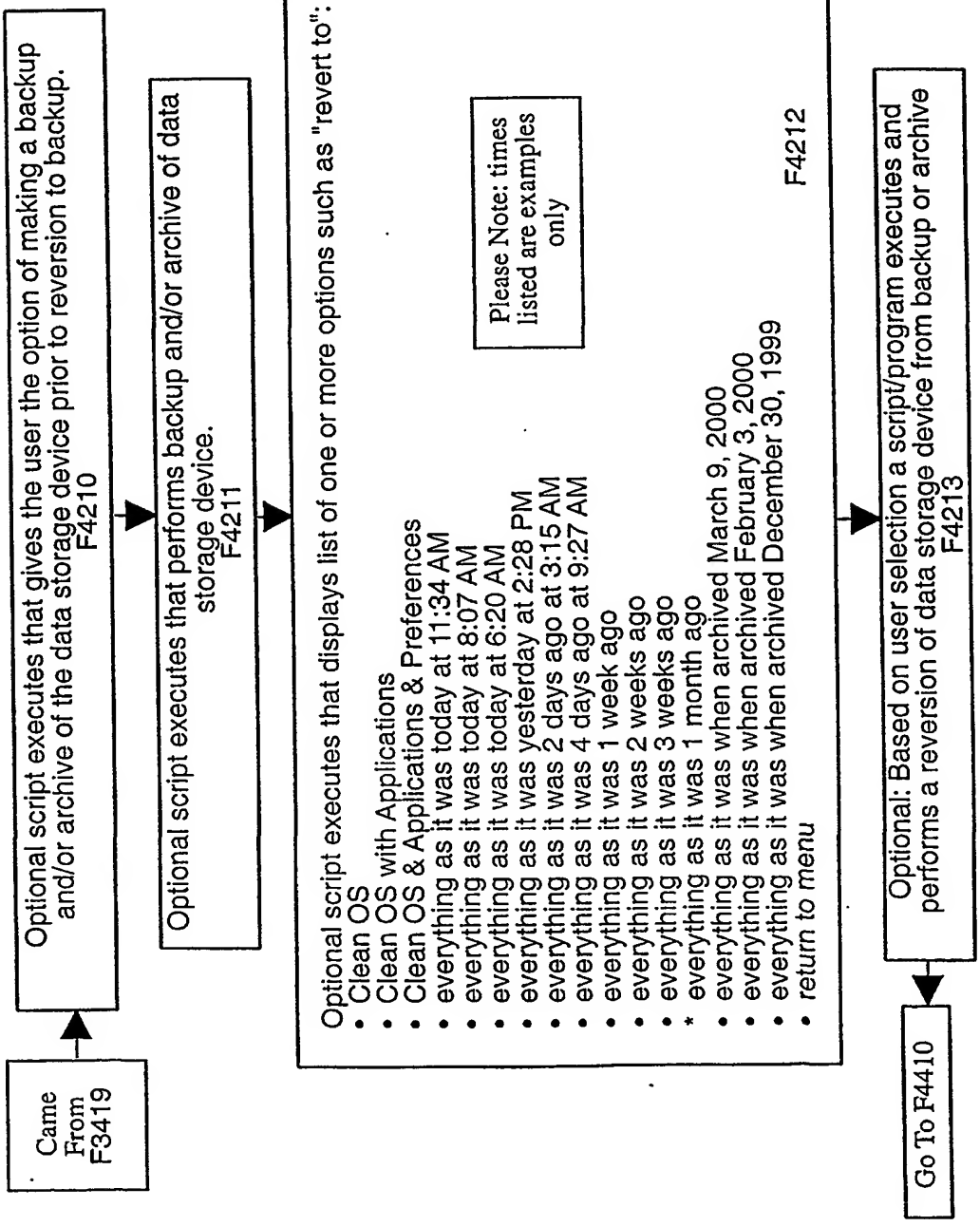

Came From F3419
Optional script executes that gives the user the option of making a backup and/or archive of the data storage device prior to reversion to backup. F4210
Optional script executes that performs backup and/or archive of data storage device. F4211
Optional script executes that displays list of one or more options such as "revert to":
• Clean OS
• Clean OS with Applications
• Clean OS & Applications & Preferences
• everything as it was today at 11:34 AM
• everything as it was today at 8:07 AM
• everything as it was today at 6:20 AM
• everything as it was yesterday at 2:28 PM
• everything as it was 2 days ago at 3:15 AM
• everything as it was 4 days ago at 9:27 AM
• everything as it was 1 week ago
• everything as it was 2 weeks ago
• everything as it was 3 weeks ago
• everything as it was 1 month ago
• everything as it was when archived March 9, 2000
• everything as it was when archived February 3, 2000
• everything as it was when archived December 30, 1999
• return to menu
F4212
Please Note: times listed are examples only
Optional: Based on user selection a script/program executes and performs a reversion of data storage device from backup or archive F4213
Go To F4410

FIG._21M

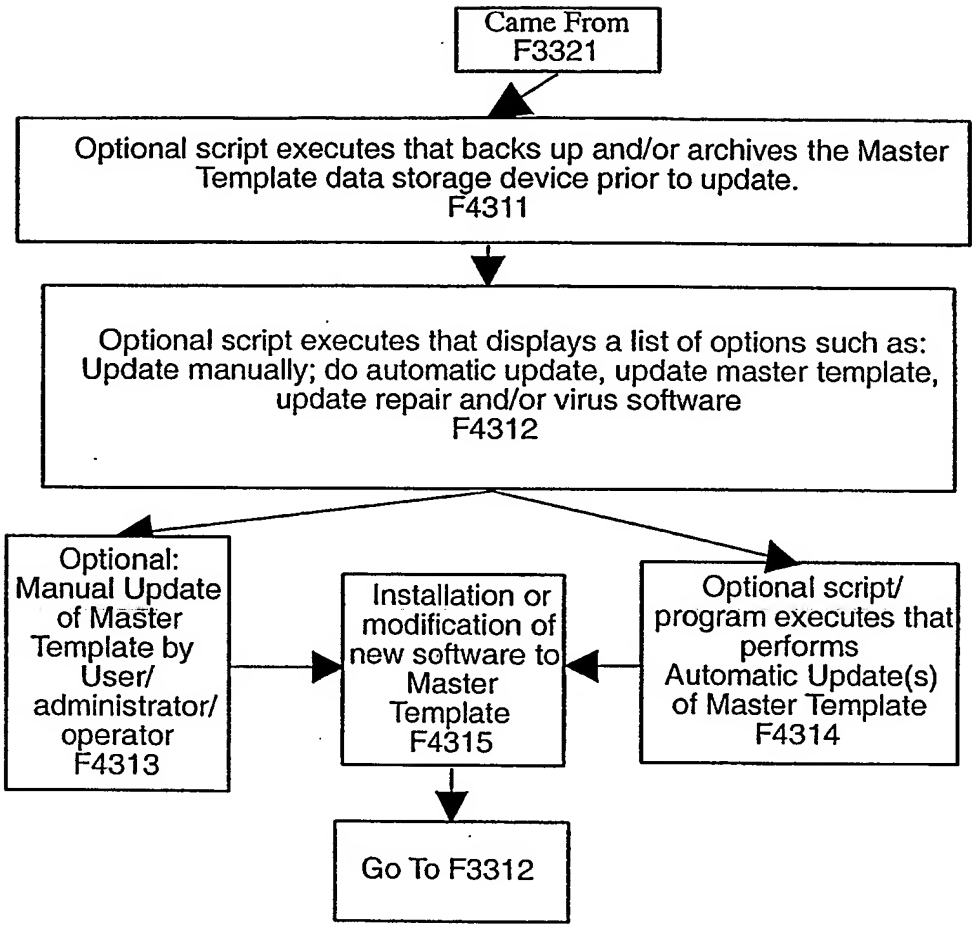

Came From F3321
Optional script executes that backs up and/or archives the Master Template data storage device prior to update. F4311
Optional script executes that displays a list of options such as: Update manually; do automatic update, update master template, update repair and/or virus software F4312
Optional: Manual Update of Master Template by User/ administrator/ operator F4313
Installation or modification of new software to Master Template F4315
Optional script/ program executes that performs Automatic Update(s) of Master Template F4314
Go To F3312

FIG._21N

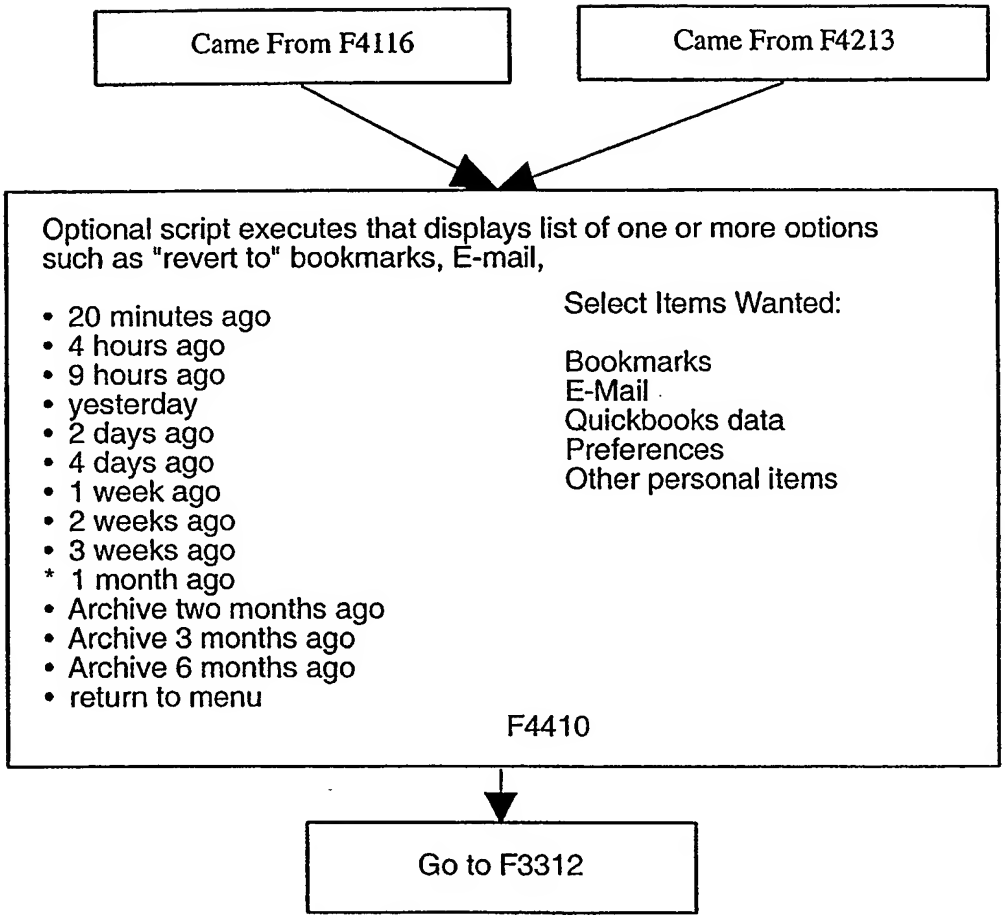

Came From F4116
Came From F4213
Optional script executes that displays list of one or more options such as "revert to" bookmarks, E-mail,
• 20 minutes ago
• 4 hours ago
• 9 hours ago
• yesterday
• 2 days ago
• 4 days ago
• 1 week ago
• 2 weeks ago
• 3 weeks ago
* 1 month ago
• Archive two months ago
• Archive 3 months ago
• Archive 6 months ago
• return to menu
Select Items Wanted:
Bookmarks
E-Mail
Quickbooks data
Preferences
Other personal items
F4410
Go to F3312

FIG._21O

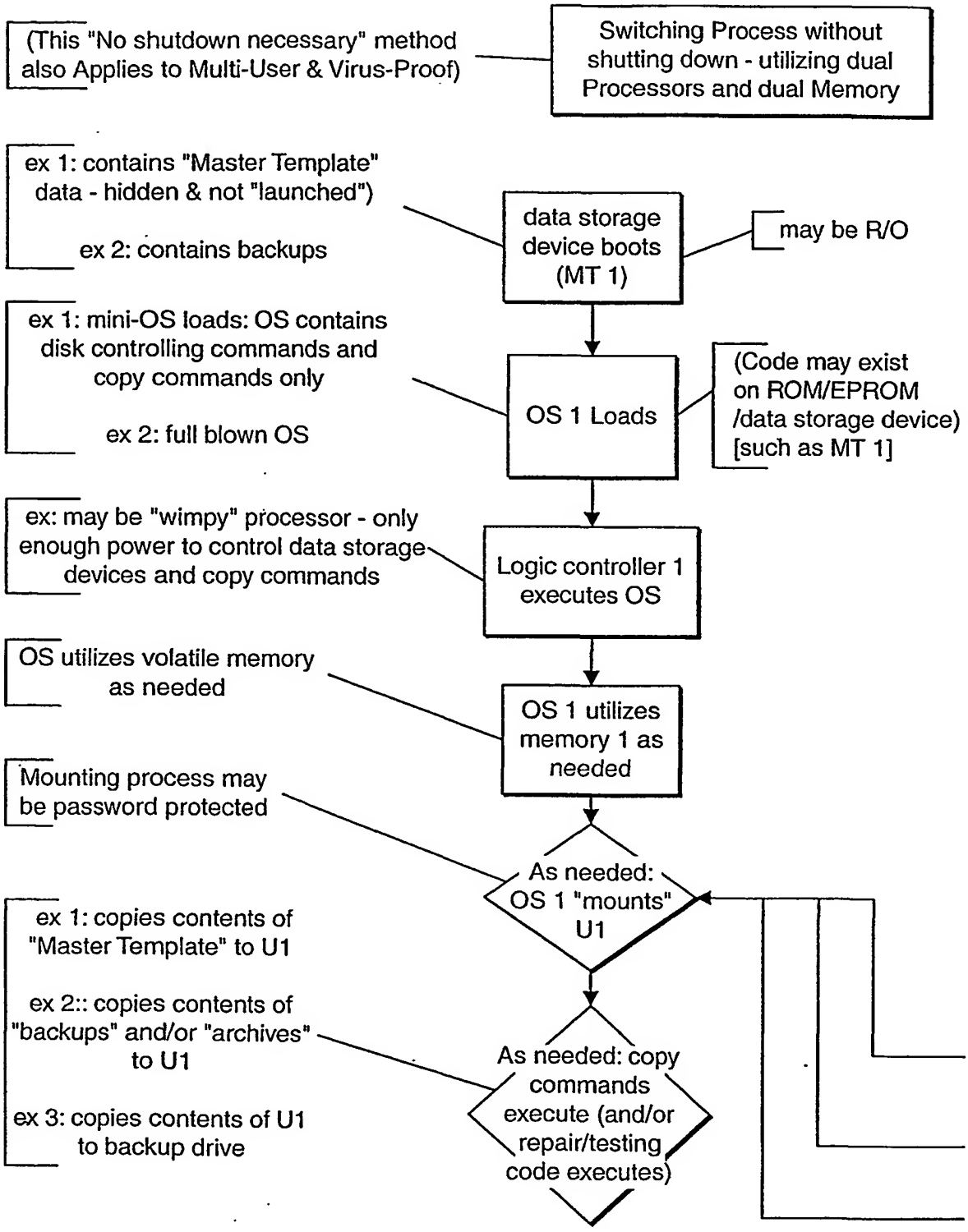

Switching Process without shutting down - utilizing dual Processors and dual Memory
(This "No shutdown necessary" method also Applies to Multi-User & Virus-Proof)
data storage device boots (MT 1)
may be R/O
ex 1: contains "Master Template" data - hidden & not "launched")
ex 2: contains backups
OS 1 Loads
(Code may exist on ROM/EPROM /data storage device) [such as MT 1]
ex 1: mini-OS loads: OS contains disk controlling commands and copy commands only
ex 2: full blown OS
Logic controller 1 executes OS
ex: may be "wimpy" processor - only enough power to control data storage devices and copy commands
OS 1 utilizes memory 1 as needed
OS utilizes volatile memory as needed
As needed: OS 1 "mounts" U1
Mounting process may be password protected
As needed: copy commands execute (and/or repair/testing code executes)
ex 1: copies contents of "Master Template" to U1
ex 2:: copies contents of "backups" and/or "archives" to U1
ex 3: copies contents of U1 to backup drive

FIG._21P-1

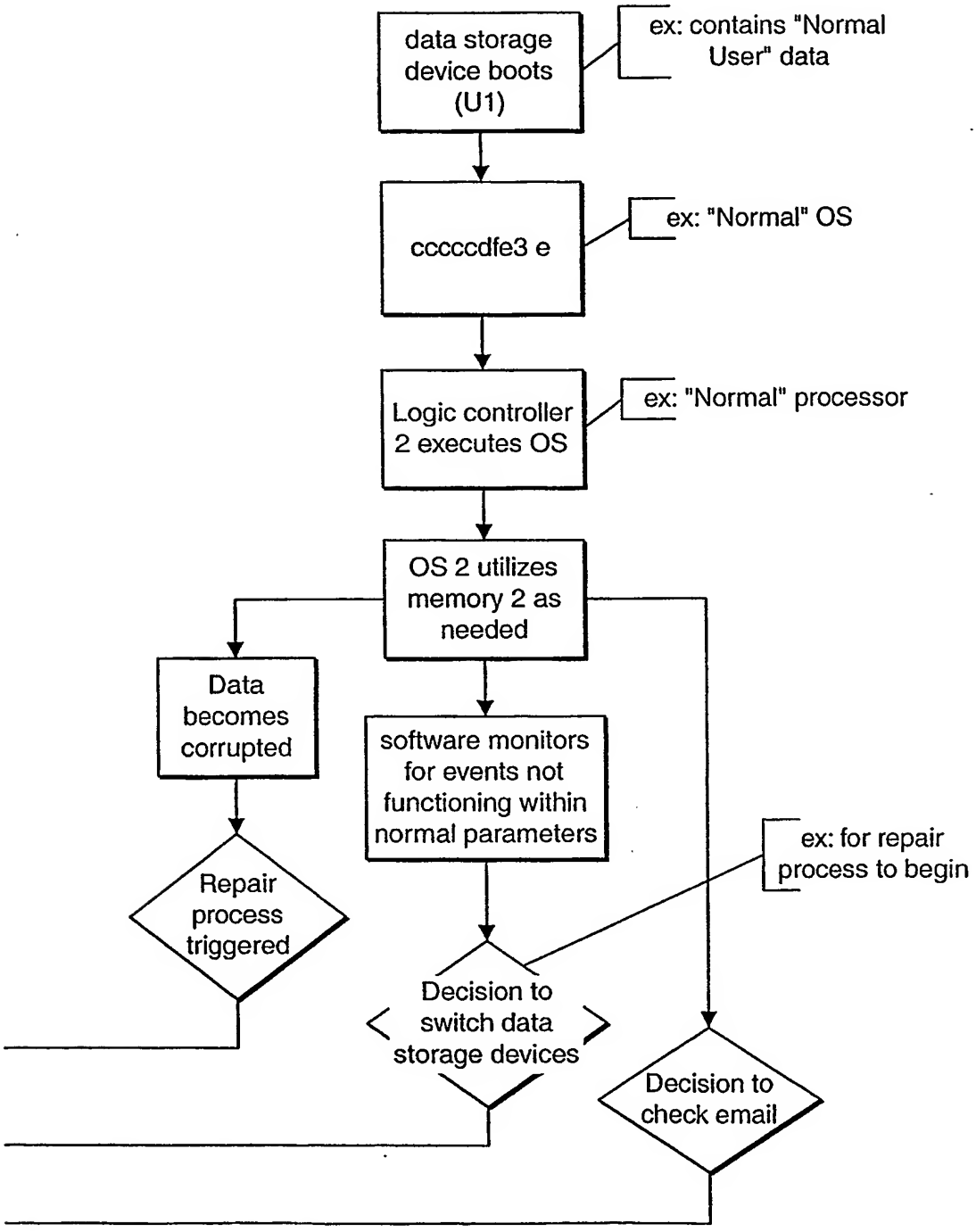
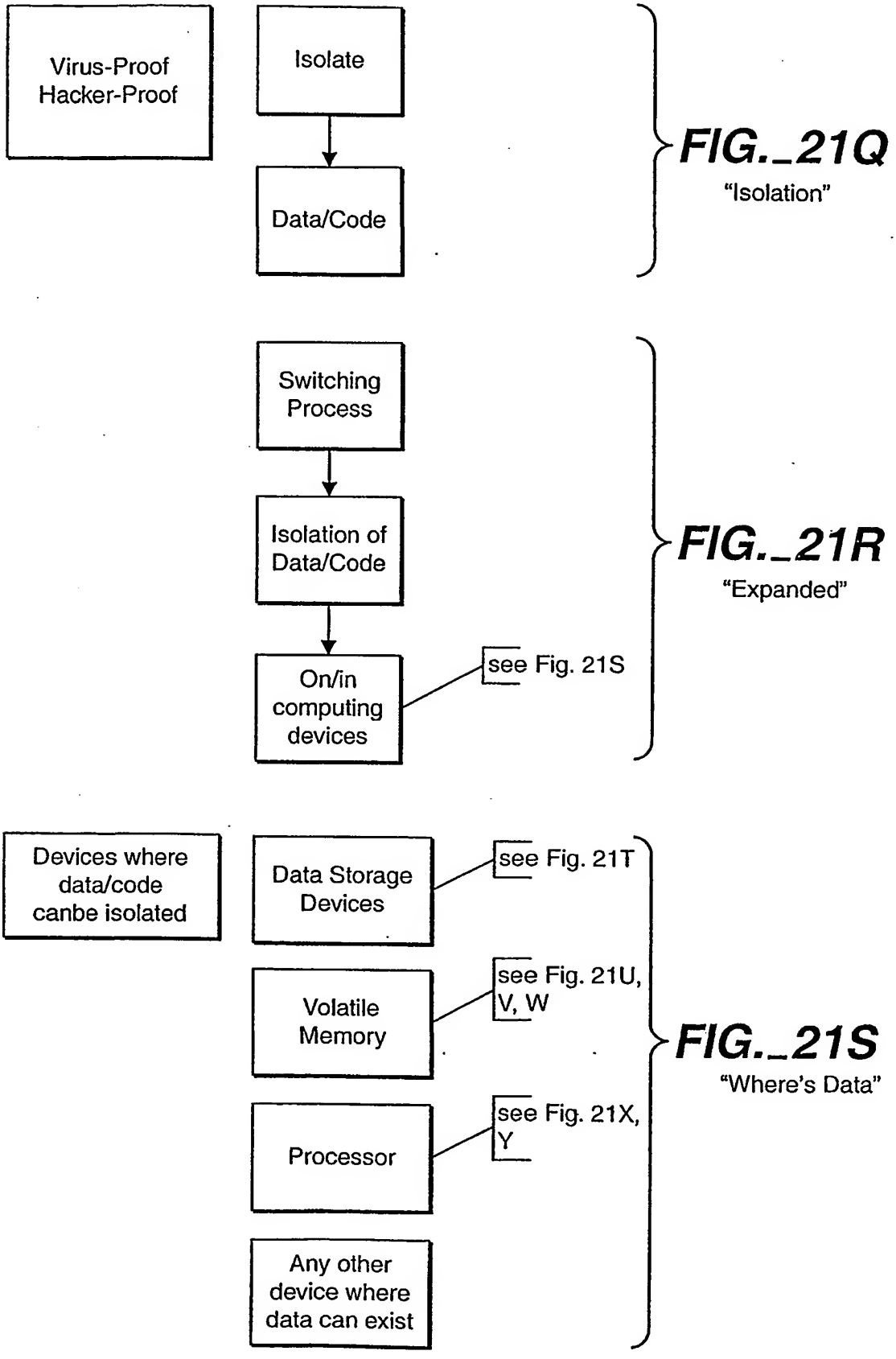


FIG._21P-2

FIG._21P

FIG._21P-1 FIG._21P-2


Virus-Proof
Hacker-Proof
Isolate
Data/Code
FIG._21Q
"Isolation"
Switching
Process
Isolation of
Data/Code
On/in
computing
devices
see Fig. 21S
FIG._21R
"Expanded"
Devices where
data/code
canbe isolated
Data Storage
Devices
see Fig. 21T
Volatile
Memory
see Fig. 21U,
V, W
Processor
see Fig. 21X,
Y
Any other
device where
data can exist
FIG._21S
"Where's Data"

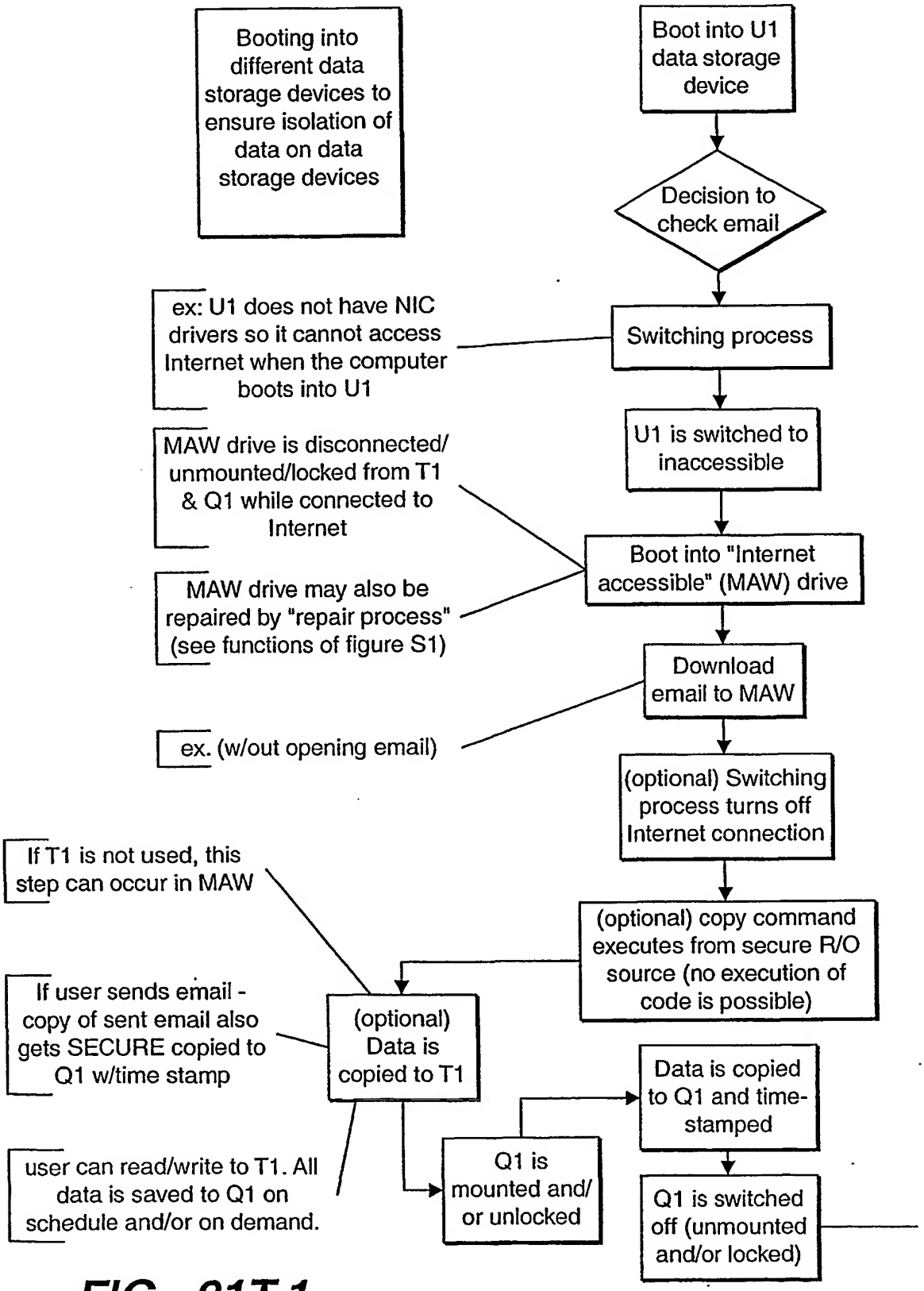

Booting into different data storage devices to ensure isolation of data on data storage devices
Boot into U1 data storage device
Decision to check email
ex: U1 does not have NIC drivers so it cannot access Internet when the computer boots into U1
Switching process
U1 is switched to inaccessible
MAW drive is disconnected/ unmounted/locked from T1 & Q1 while connected to Internet
MAW drive may also be repaired by "repair process" (see functions of figure S1)
Boot into "Internet accessible" (MAW) drive
Download email to MAW
ex. (w/out opening email)
(optional) Switching process turns off Internet connection
If T1 is not used, this step can occur in MAW
(optional) copy command executes from secure R/O source (no execution of code is possible)
If user sends email - copy of sent email also gets SECURE copied to Q1 w/time stamp
(optional) Data is copied to T1
Data is copied to Q1 and time-stamped
user can read/write to T1. All data is saved to Q1 on schedule and/or on demand.
Q1 is mounted and/ or unlocked
Q1 is switched off (unmounted and/or locked)

FIG._21T-1

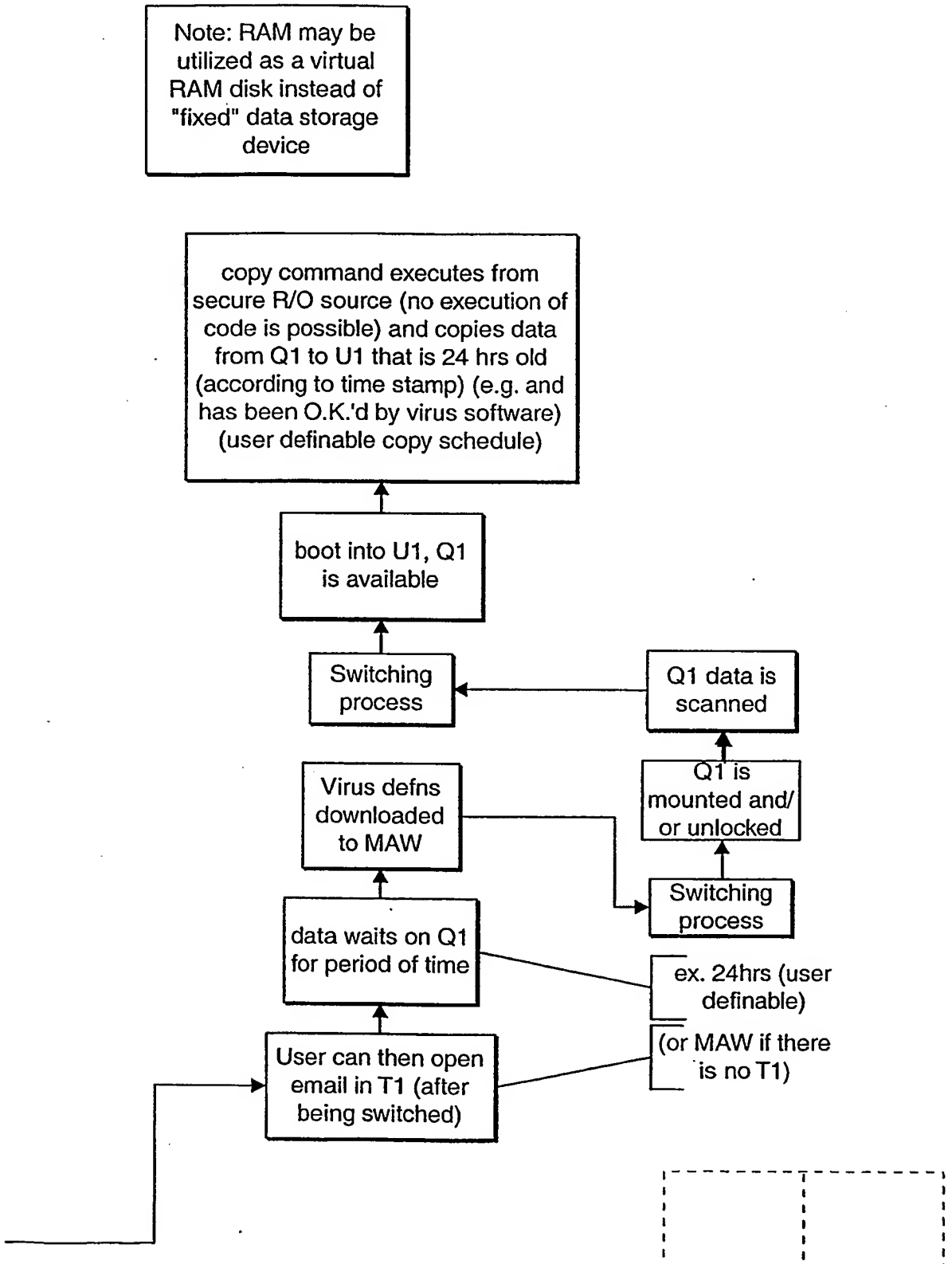


FIG._21T-2

FIG._21T

FIG._21T-1 FIG._21T-2

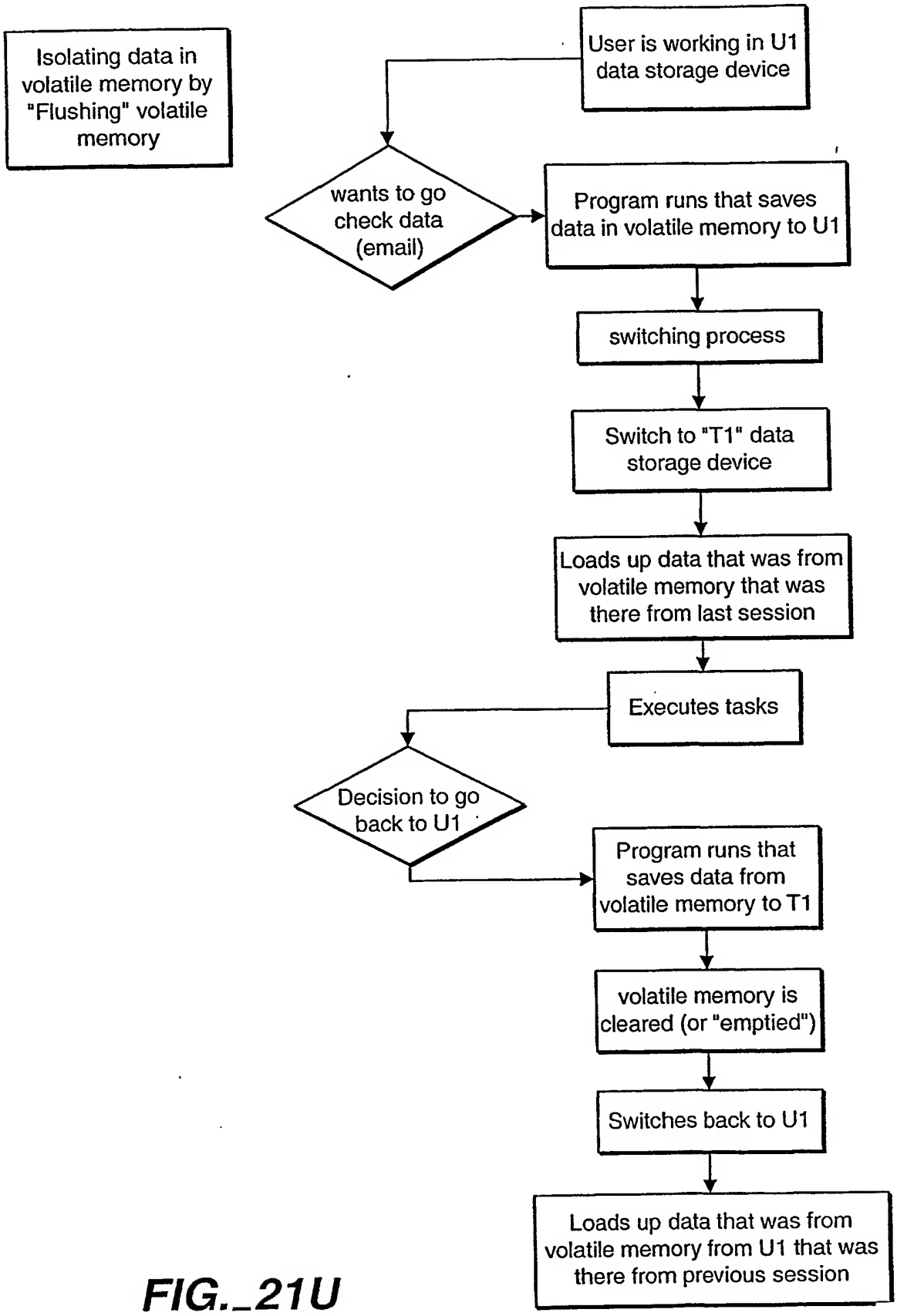

Isolating data in volatile memory by "Flushing" volatile memory
User is working in U1 data storage device
wants to go check data (email)
Program runs that saves data in volatile memory to U1
switching process
Switch to "T1" data storage device
Loads up data that was from volatile memory that was there from last session
Executes tasks
Decision to go back to U1
Program runs that saves data from volatile memory to T1
volatile memory is cleared (or "emptied")
Switches back to U1
Loads up data that was from volatile memory from U1 that was there from previous session

FIG._21U

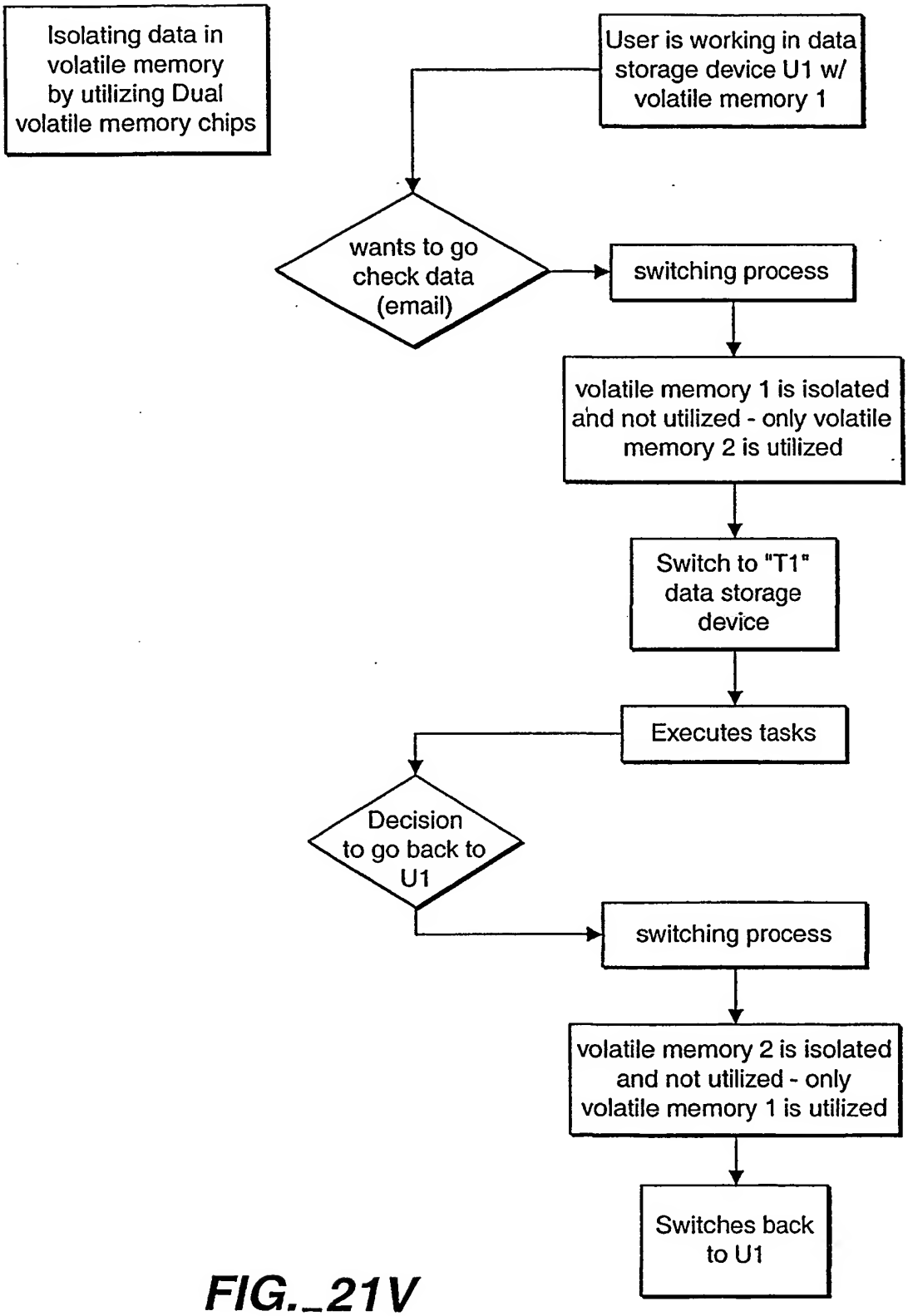

Isolating data in volatile memory by utilizing Dual volatile memory chips
User is working in data storage device U1 w/ volatile memory 1
wants to go check data (email)
switching process
volatile memory 1 is isolated and not utilized - only volatile memory 2 is utilized
Switch to "T1" data storage device
Executes tasks
Decision to go back to U1
switching process
volatile memory 2 is isolated and not utilized - only volatile memory 1 is utilized
Switches back to U1

FIG._21V

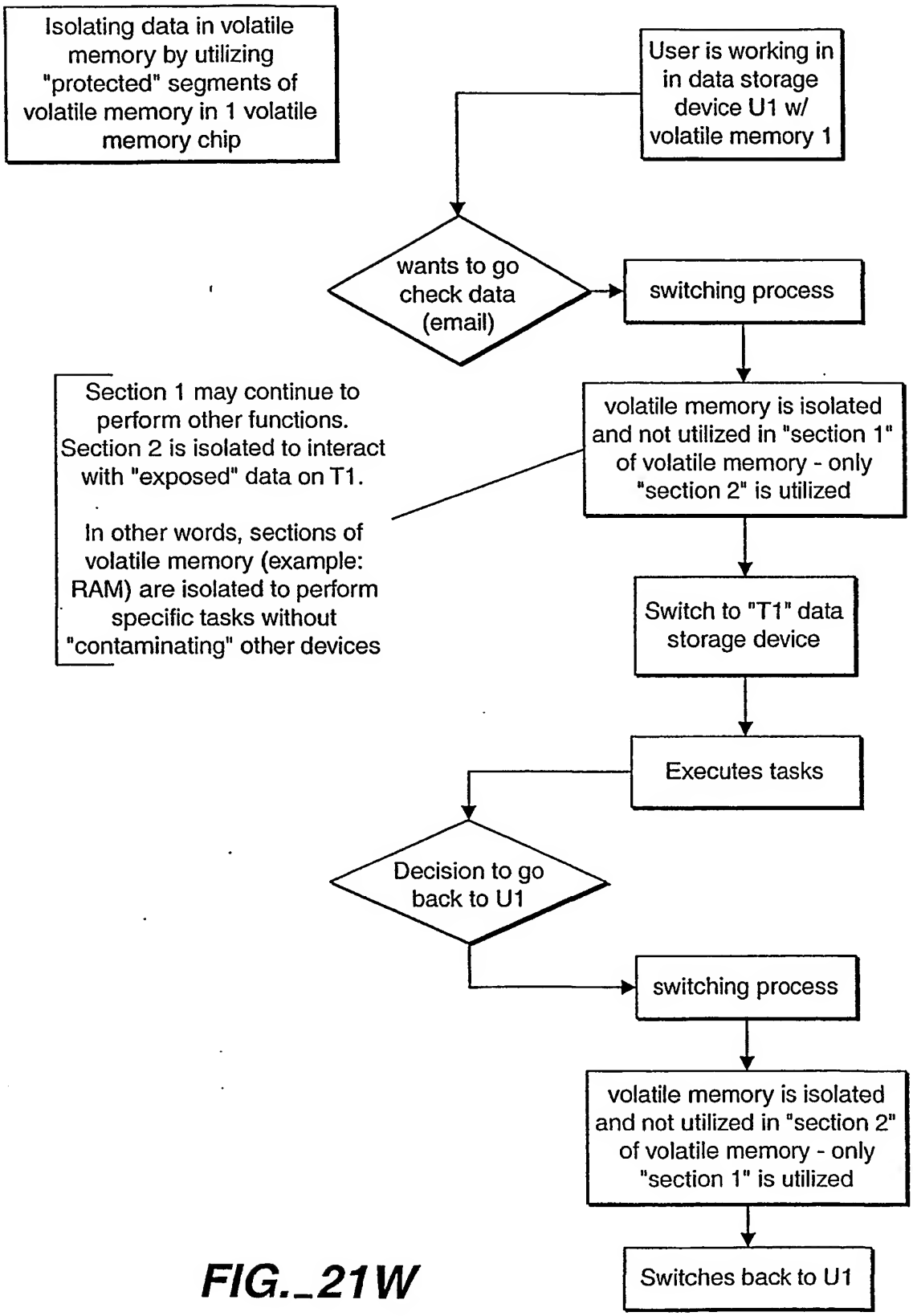

Isolating data in volatile memory by utilizing "protected" segments of volatile memory in 1 volatile memory chip
User is working in in data storage device U1 w/ volatile memory 1
wants to go check data (email)
switching process
Section 1 may continue to perform other functions. Section 2 is isolated to interact with "exposed" data on T1.
In other words, sections of volatile memory (example: RAM) are isolated to perform specific tasks without "contaminating" other devices
volatile memory is isolated and not utilized in "section 1" of volatile memory - only "section 2" is utilized
Switch to "T1" data storage device
Executes tasks
Decision to go back to U1
switching process
volatile memory is isolated and not utilized in "section 2" of volatile memory - only "section 1" is utilized
Switches back to U1

FIG._21W

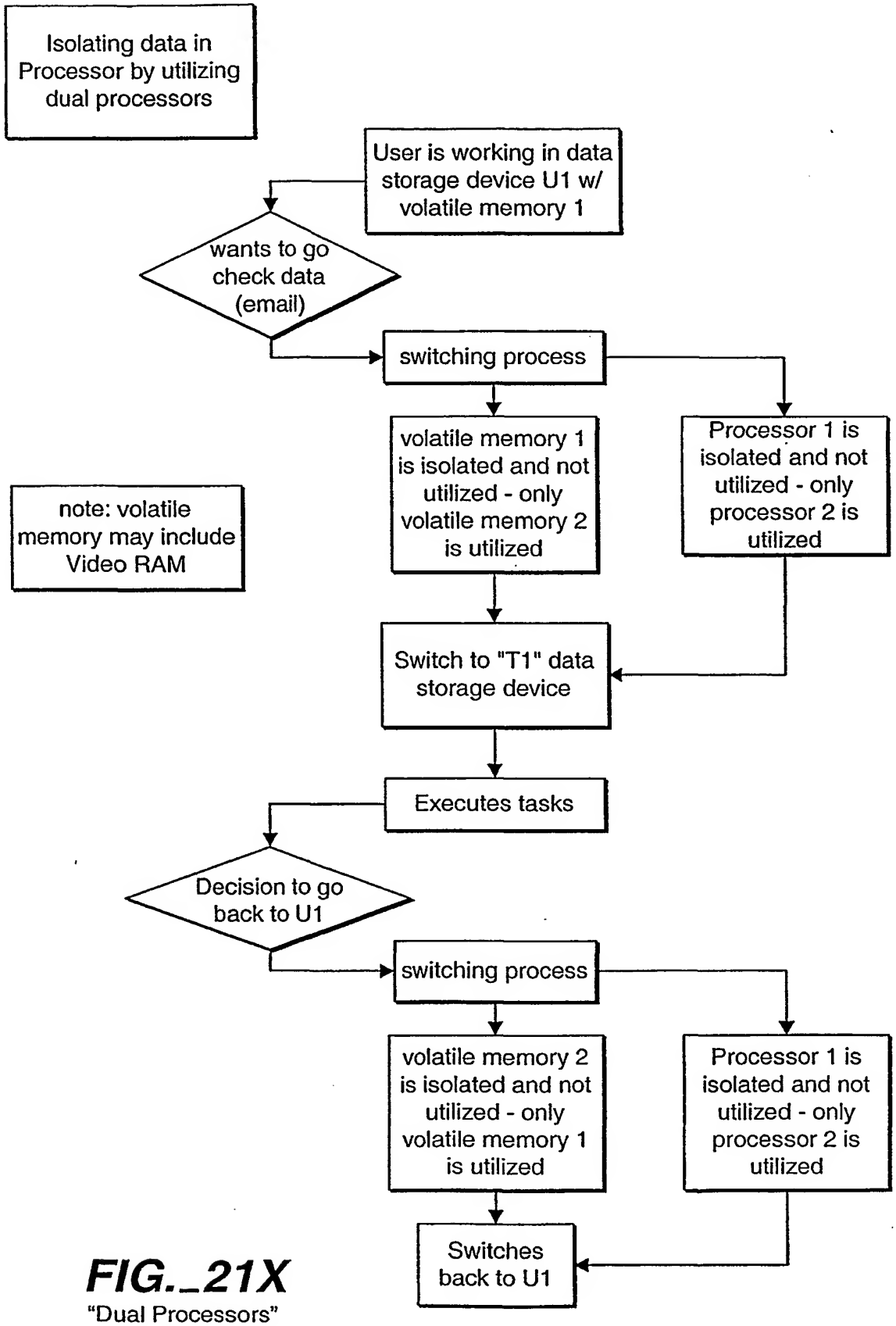


FIG._21X
"Dual Processors"

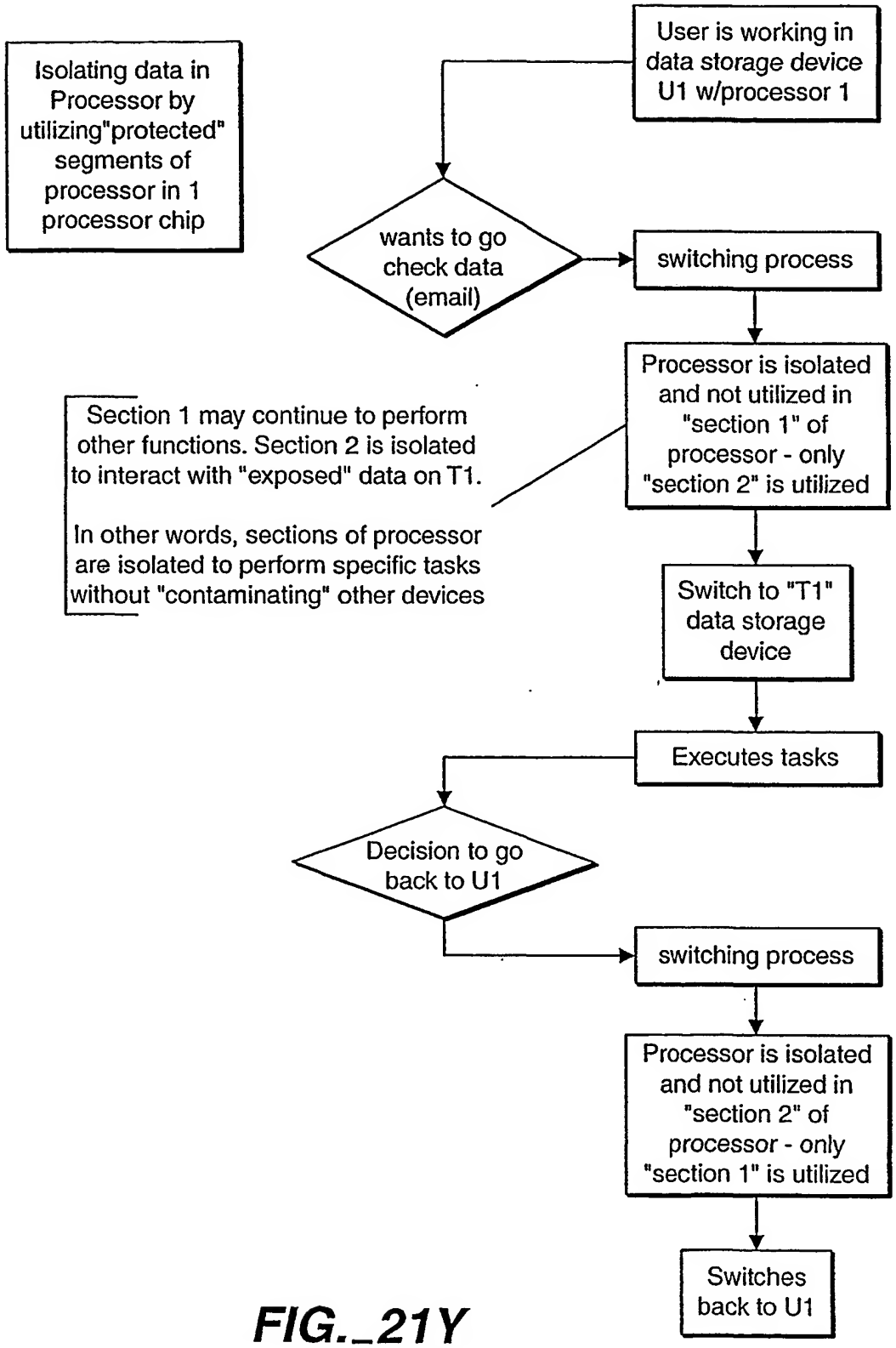


FIG._21Y

"Protected"

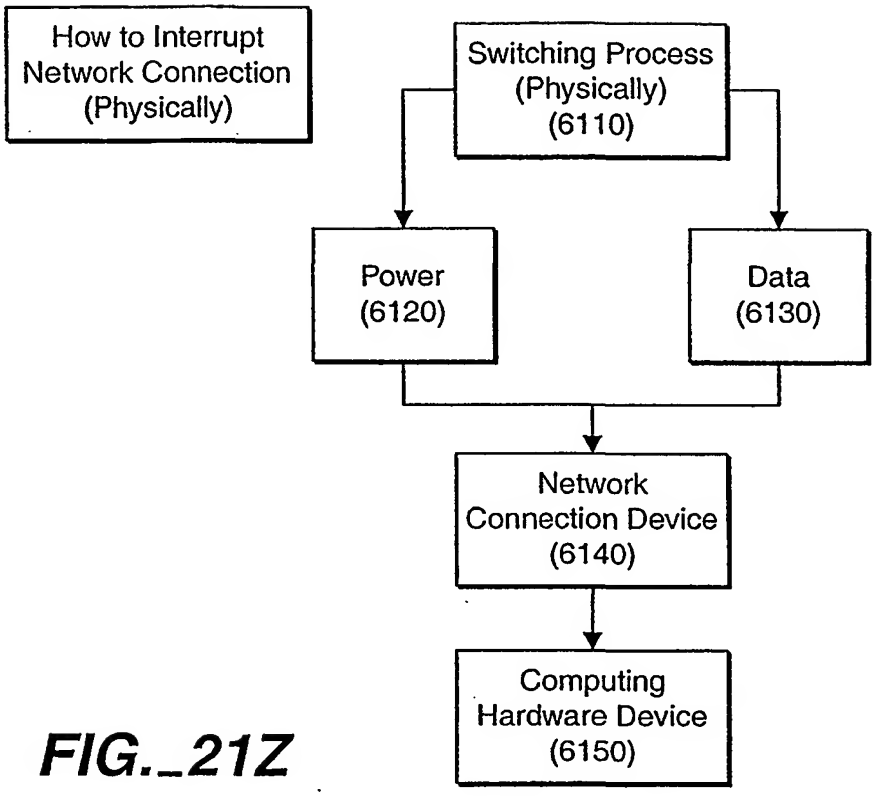


FIG._21Z

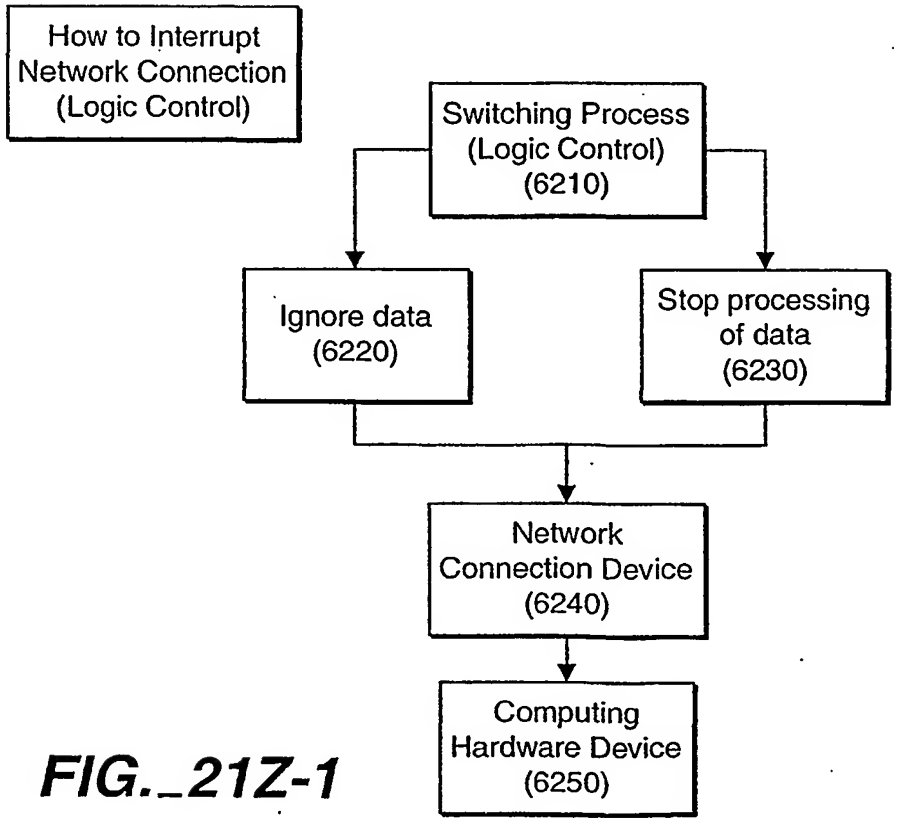


FIG._21Z-1

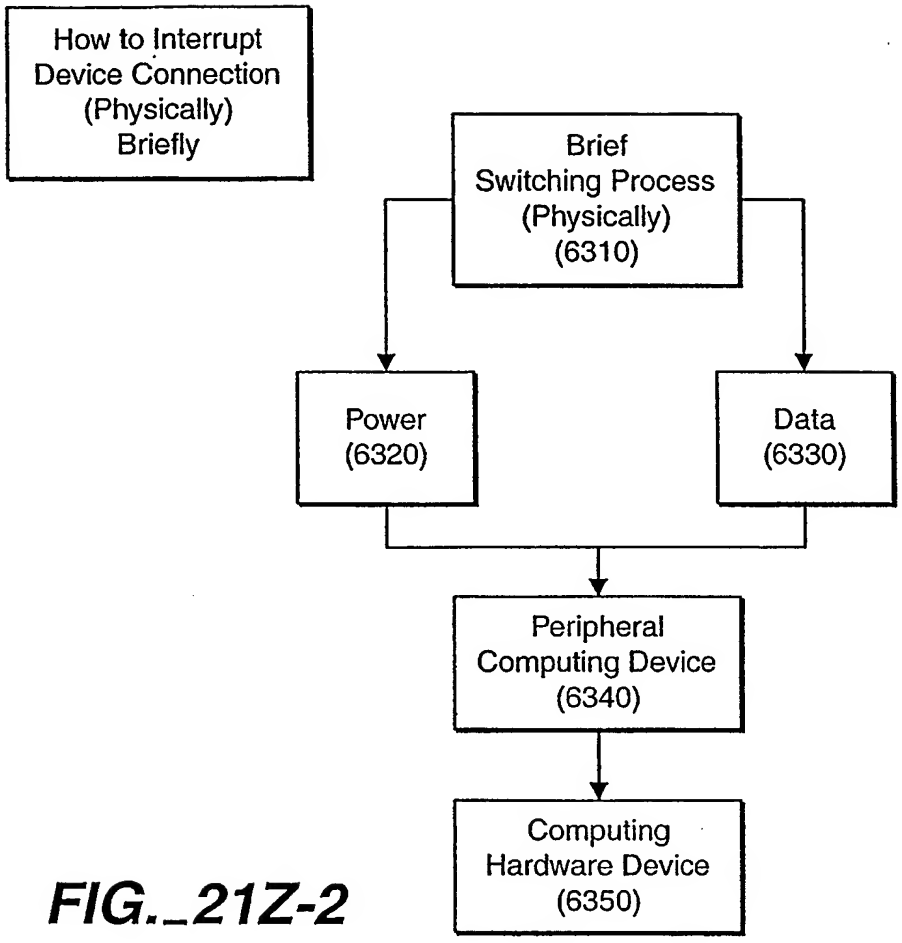

How to Interrupt Device Connection (Physically) Briefly
Brief Switching Process (Physically) (6310)
Power (6320)
Data (6330)
Peripheral Computing Device (6340)
Computing Hardware Device (6350)

FIG._21Z-2

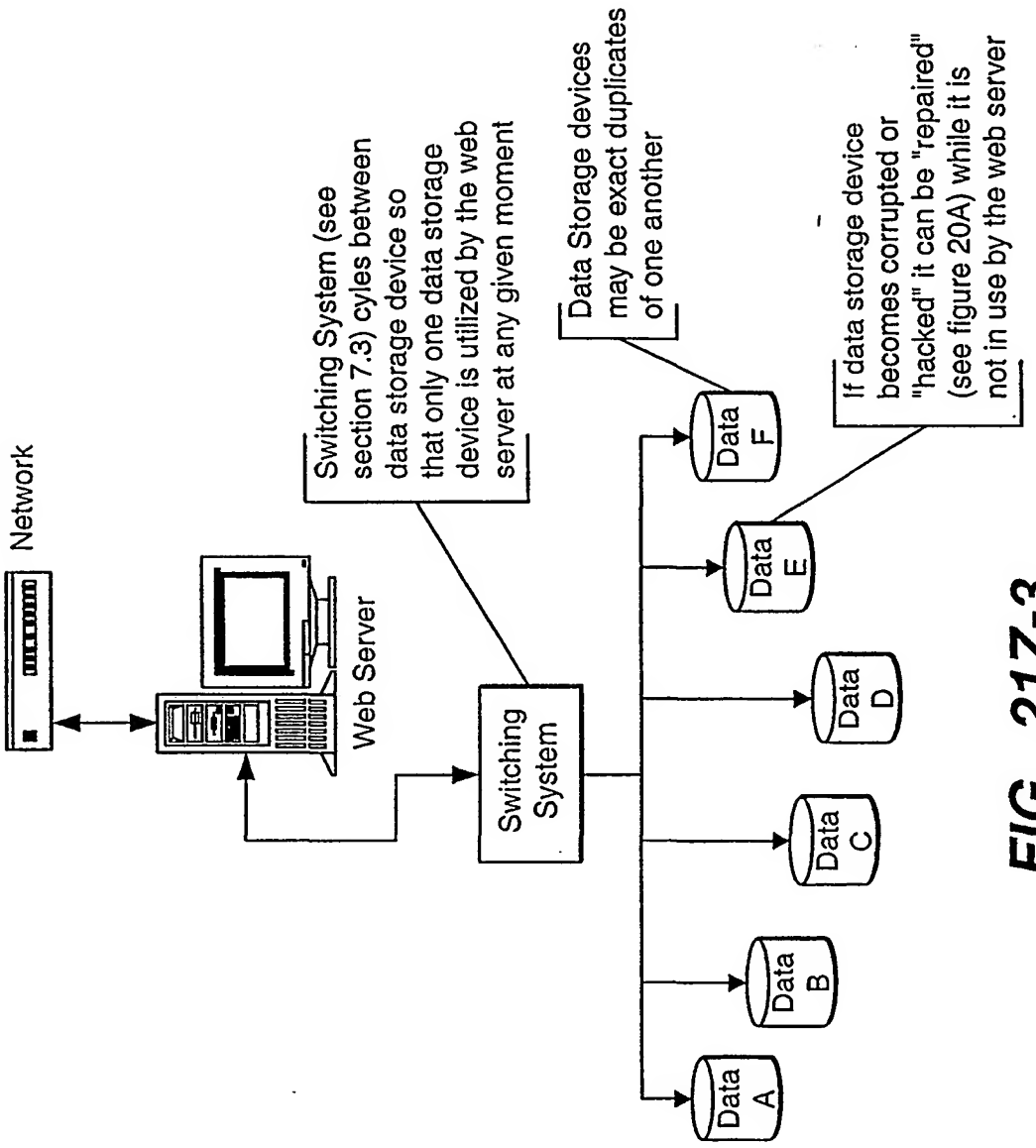

Network
Web Server
Switching System
Switching System (see section 7.3) cyles between data storage device so that only one data storage device is utilized by the web server at any given moment
Data Storage devices may be exact duplicates of one another
If data storage device becomes corrupted or "hacked" it can be "repaired" (see figure 20A) while it is not in use by the web server
Data A
Data B
Data C
Data D
Data E
Data F

FIG._21Z-3

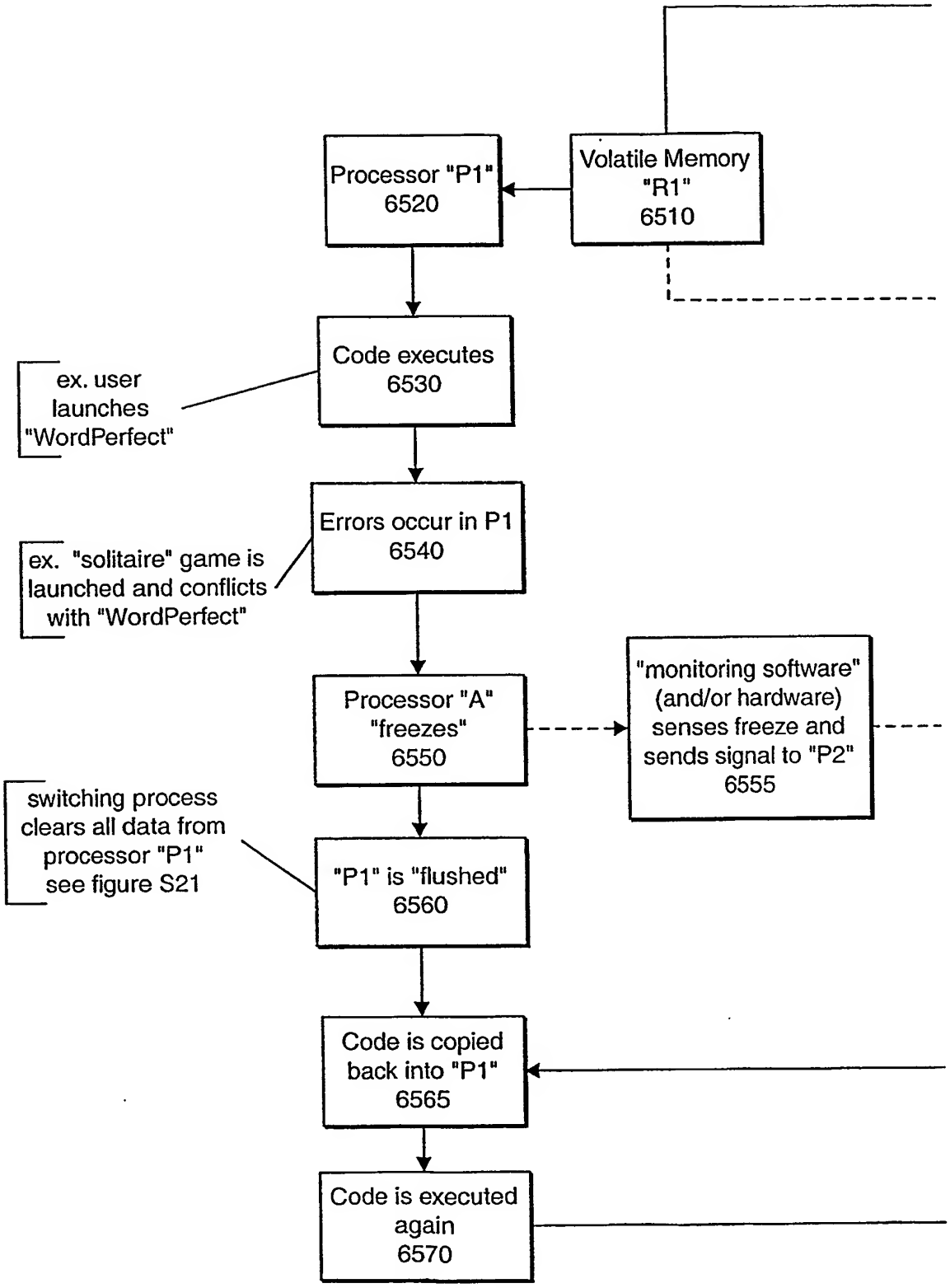

Processor "P1"
6520
Volatile Memory
"R1"
6510
Code executes
6530
ex. user
launches
"WordPerfect"
Errors occur in P1
6540
ex. "solitaire" game is
launched and conflicts
with "WordPerfect"
Processor "A"
"freezes"
6550
"monitoring software"
(and/or hardware)
senses freeze and
sends signal to "P2"
6555
switching process
clears all data from
processor "P1"
see figure S21
"P1" is "flushed"
6560
Code is copied
back into "P1"
6565
Code is executed
again
6570

FIG._21Z-4a

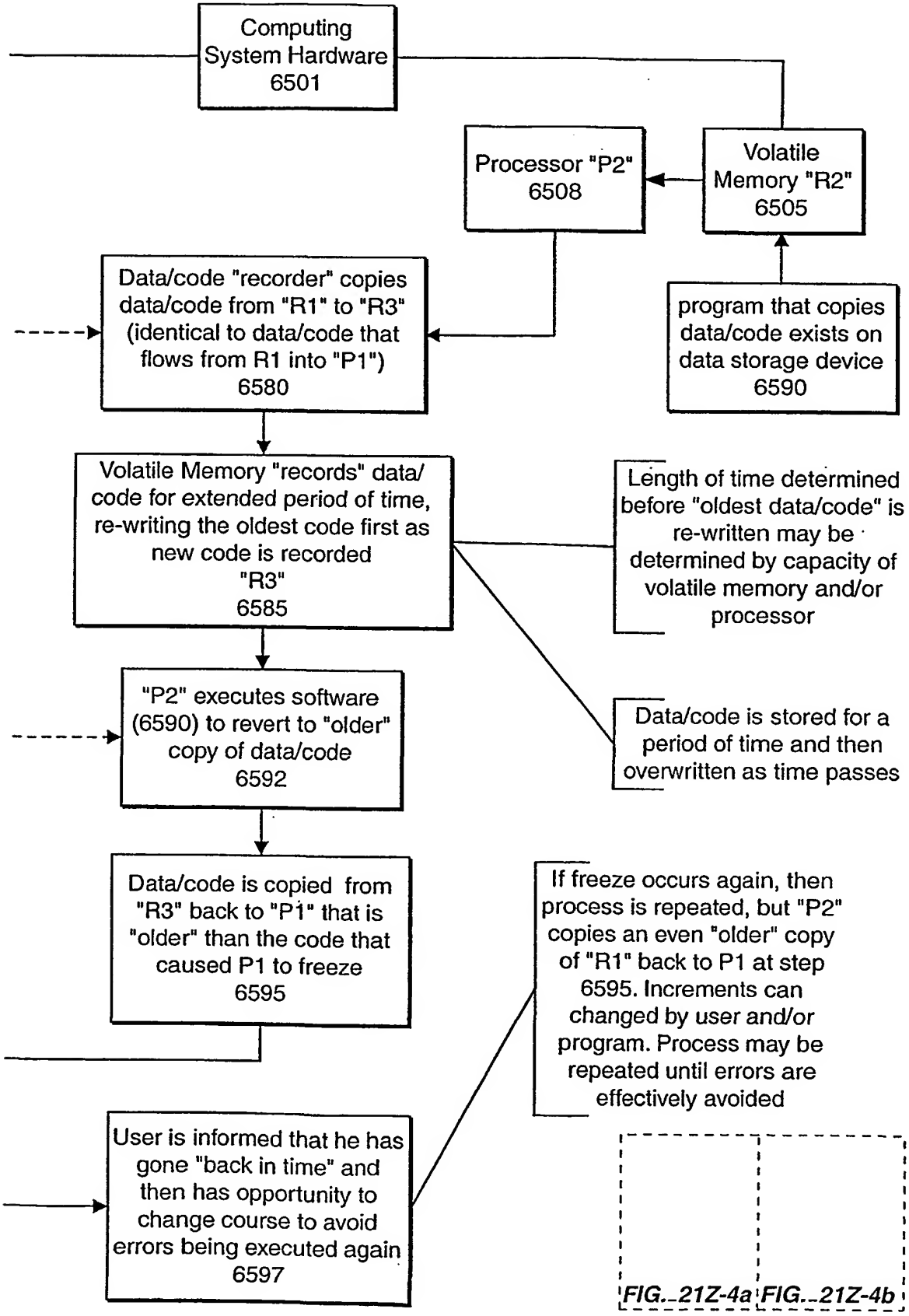


FIG._21Z-4b

FIG._21Z-4a FIG._21Z-4b

FIG._21Z-4

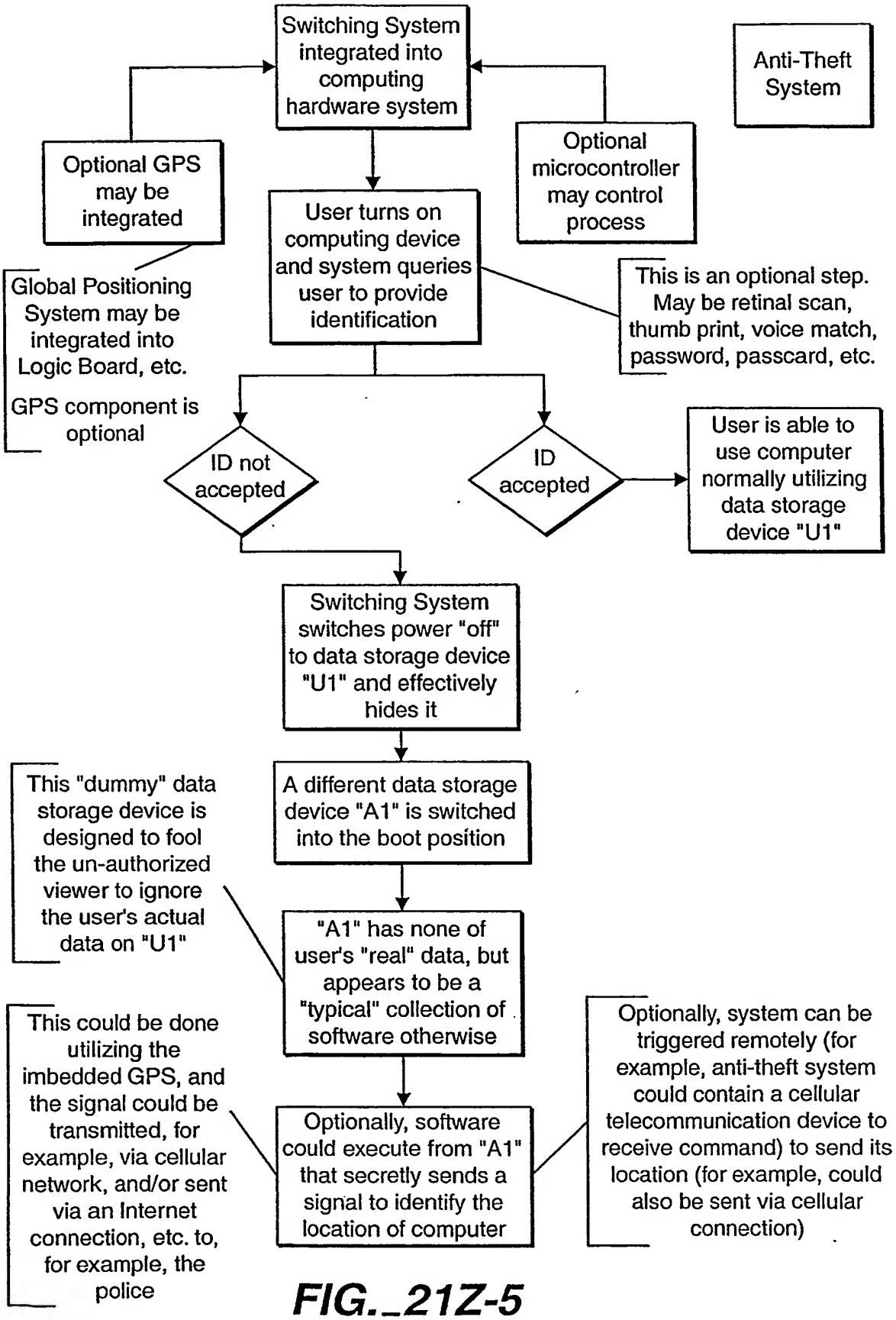

Switching System integrated into computing hardware system
Anti-Theft System
Optional GPS may be integrated
Optional microcontroller may control process
User turns on computing device and system queries user to provide identification
Global Positioning System may be integrated into Logic Board, etc.
GPS component is optional
This is an optional step. May be retinal scan, thumb print, voice match, password, passcard, etc.
ID not accepted
ID accepted
User is able to use computer normally utilizing data storage device "U1"
Switching System switches power "off" to data storage device "U1" and effectively hides it
A different data storage device "A1" is switched into the boot position
This "dummy" data storage device is designed to fool the un-authorized viewer to ignore the user's actual data on "U1"
"A1" has none of user's "real" data, but appears to be a "typical" collection of software otherwise
This could be done utilizing the imbedded GPS, and the signal could be transmitted, for example, via cellular network, and/or sent via an Internet connection, etc. to, for example, the police
Optionally, software could execute from "A1" that secretly sends a signal to identify the location of computer
Optionally, system can be triggered remotely (for example, anti-theft system could contain a cellular telecommunication device to receive command) to send its location (for example, could also be sent via cellular connection)

FIG._21Z-5

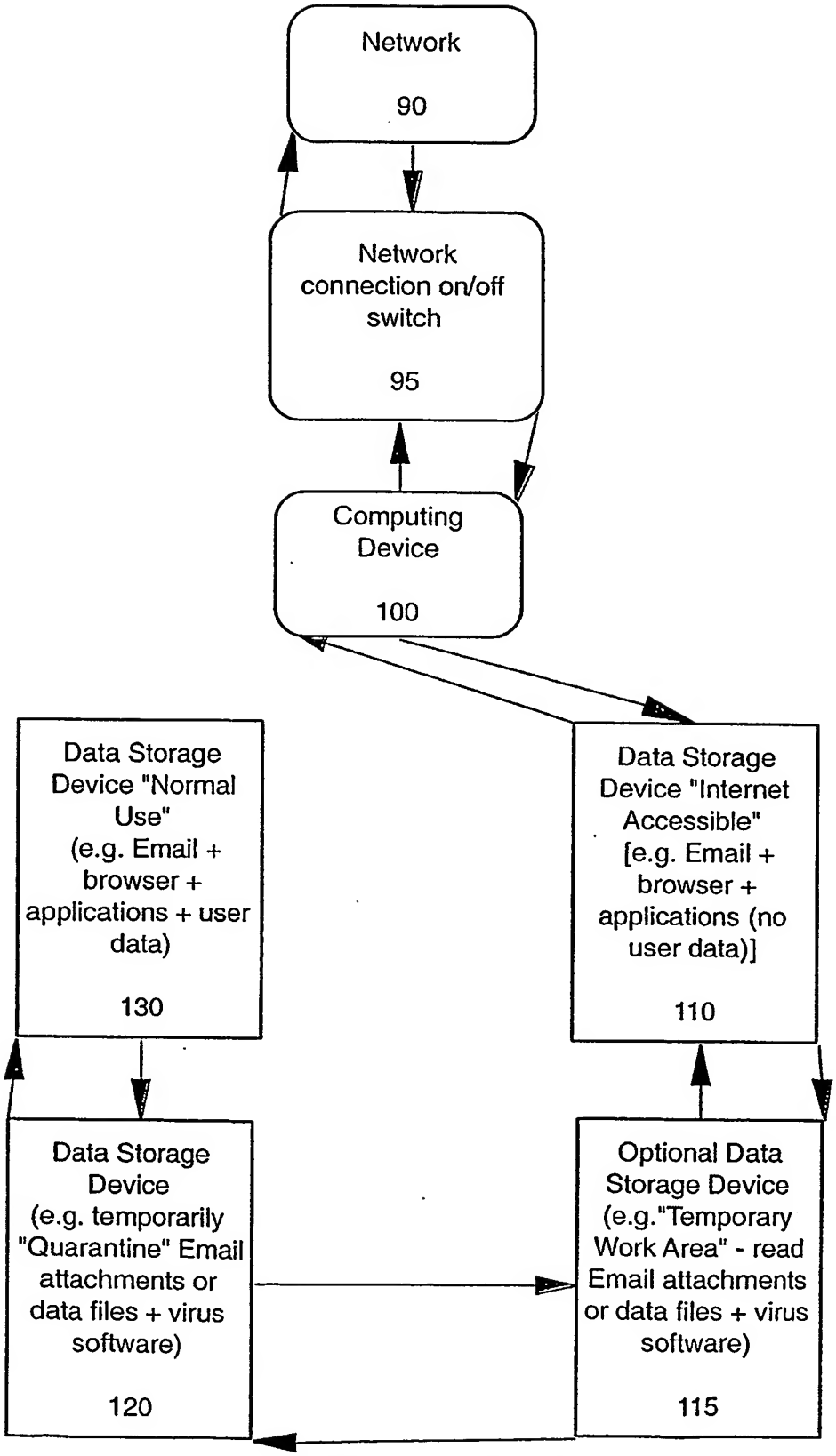

Network
90
Network connection on/off switch
95
Computing Device
100
Data Storage Device "Normal Use" (e.g. Email + browser + applications + user data)
130
Data Storage Device "Internet Accessible" [e.g. Email + browser + applications (no user data)]
110
Data Storage Device (e.g. temporarily "Quarantine" Email attachments or data files + virus software)
120
Optional Data Storage Device (e.g."Temporary Work Area" - read Email attachments or data files + virus software)
115

FIG._21Z-6

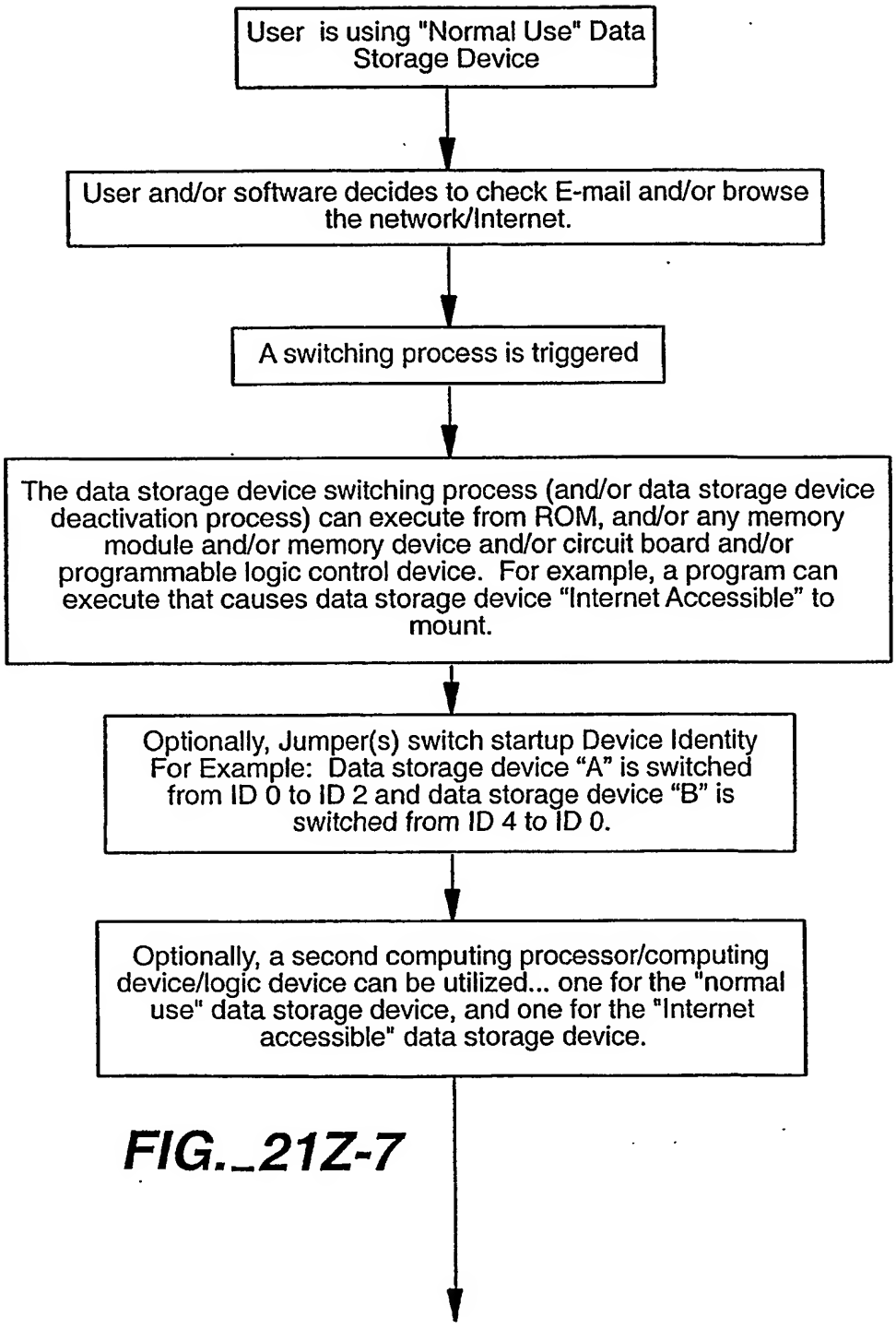

User is using "Normal Use" Data Storage Device
User and/or software decides to check E-mail and/or browse the network/Internet.
A switching process is triggered
The data storage device switching process (and/or data storage device deactivation process) can execute from ROM, and/or any memory module and/or memory device and/or circuit board and/or programmable logic control device. For example, a program can execute that causes data storage device "Internet Accessible" to mount.
Optionally, Jumper(s) switch startup Device Identity For Example: Data storage device "A" is switched from ID 0 to ID 2 and data storage device "B" is switched from ID 4 to ID 0.
Optionally, a second computing processor/computing device/logic device can be utilized... one for the "normal use" data storage device, and one for the "Internet accessible" data storage device.

FIG._21Z-7

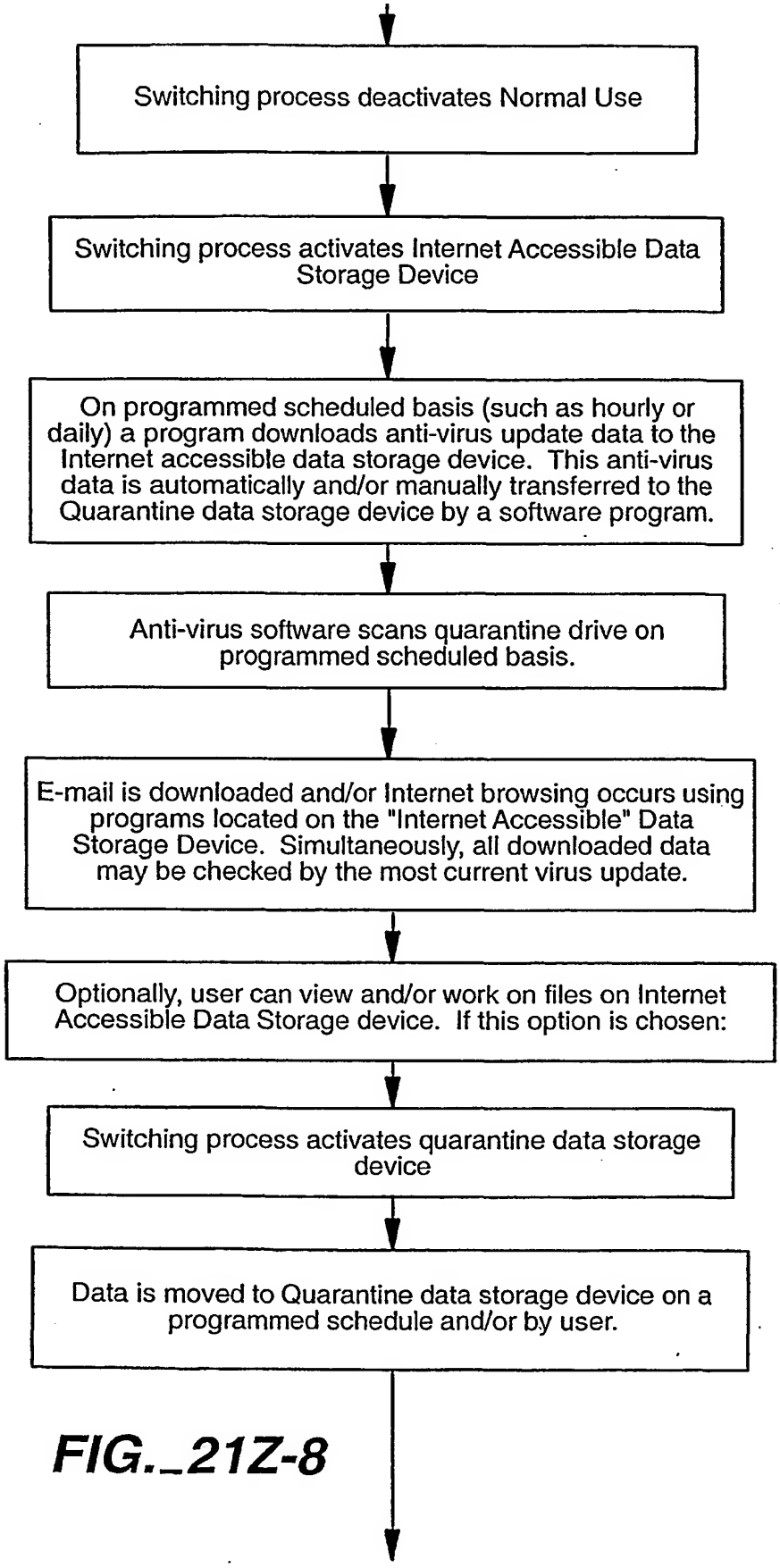

Switching process deactivates Normal Use
Switching process activates Internet Accessible Data Storage Device
On programmed scheduled basis (such as hourly or daily) a program downloads anti-virus update data to the Internet accessible data storage device. This anti-virus data is automatically and/or manually transferred to the Quarantine data storage device by a software program.
Anti-virus software scans quarantine drive on programmed scheduled basis.
E-mail is downloaded and/or Internet browsing occurs using programs located on the "Internet Accessible" Data Storage Device. Simultaneously, all downloaded data may be checked by the most current virus update.
Optionally, user can view and/or work on files on Internet Accessible Data Storage device. If this option is chosen:
Switching process activates quarantine data storage device
Data is moved to Quarantine data storage device on a programmed schedule and/or by user.

FIG._21Z-8

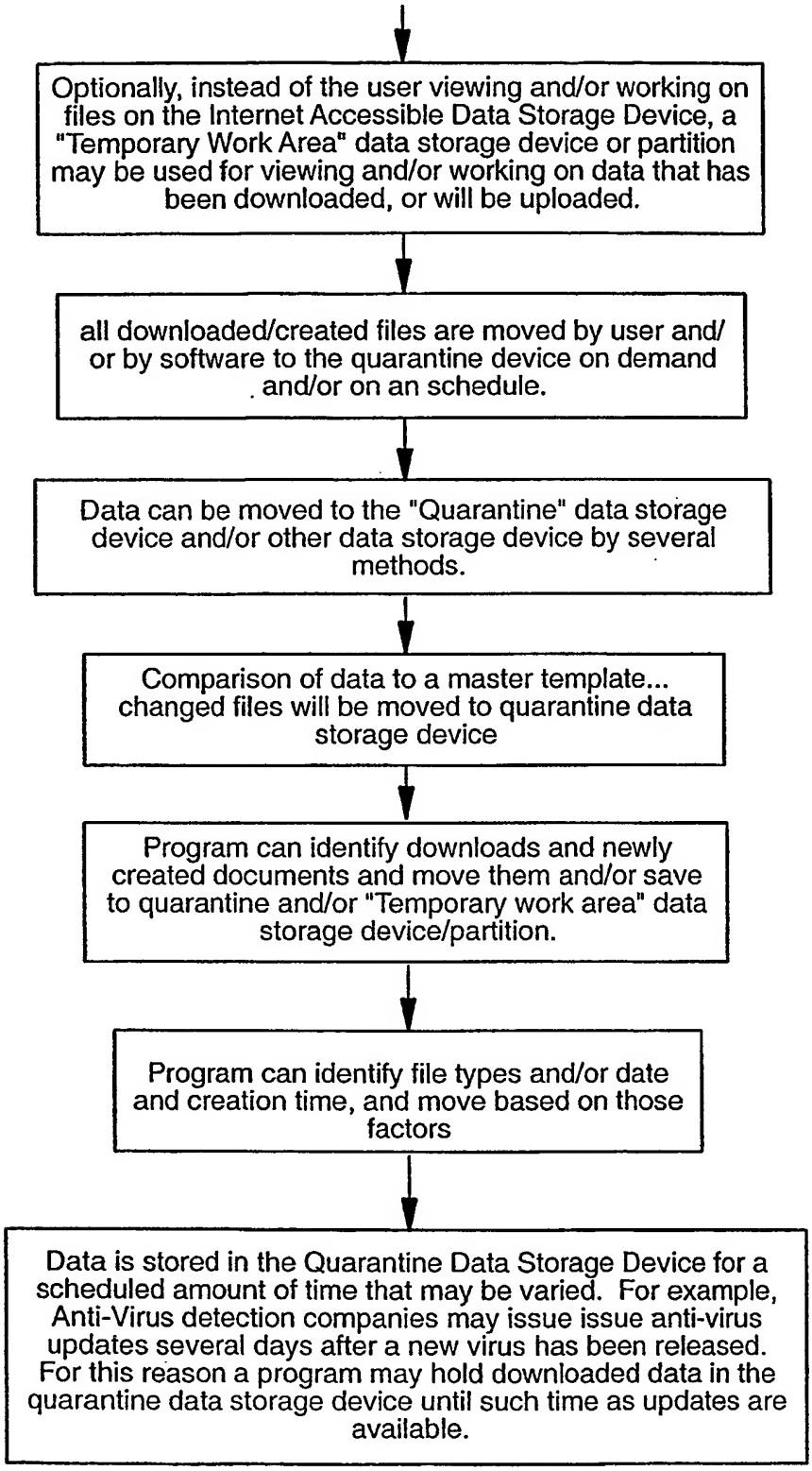

Optionally, instead of the user viewing and/or working on files on the Internet Accessible Data Storage Device, a "Temporary Work Area" data storage device or partition may be used for viewing and/or working on data that has been downloaded, or will be uploaded.
all downloaded/created files are moved by user and/ or by software to the quarantine device on demand and/or on an schedule.
Data can be moved to the "Quarantine" data storage device and/or other data storage device by several methods.
Comparison of data to a master template... changed files will be moved to quarantine data storage device
Program can identify downloads and newly created documents and move them and/or save to quarantine and/or "Temporary work area" data storage device/partition.
Program can identify file types and/or date and creation time, and move based on those factors
Data is stored in the Quarantine Data Storage Device for a scheduled amount of time that may be varied. For example, Anti-Virus detection companies may issue issue anti-virus updates several days after a new virus has been released. For this reason a program may hold downloaded data in the quarantine data storage device until such time as updates are available.

FIG._21Z-9

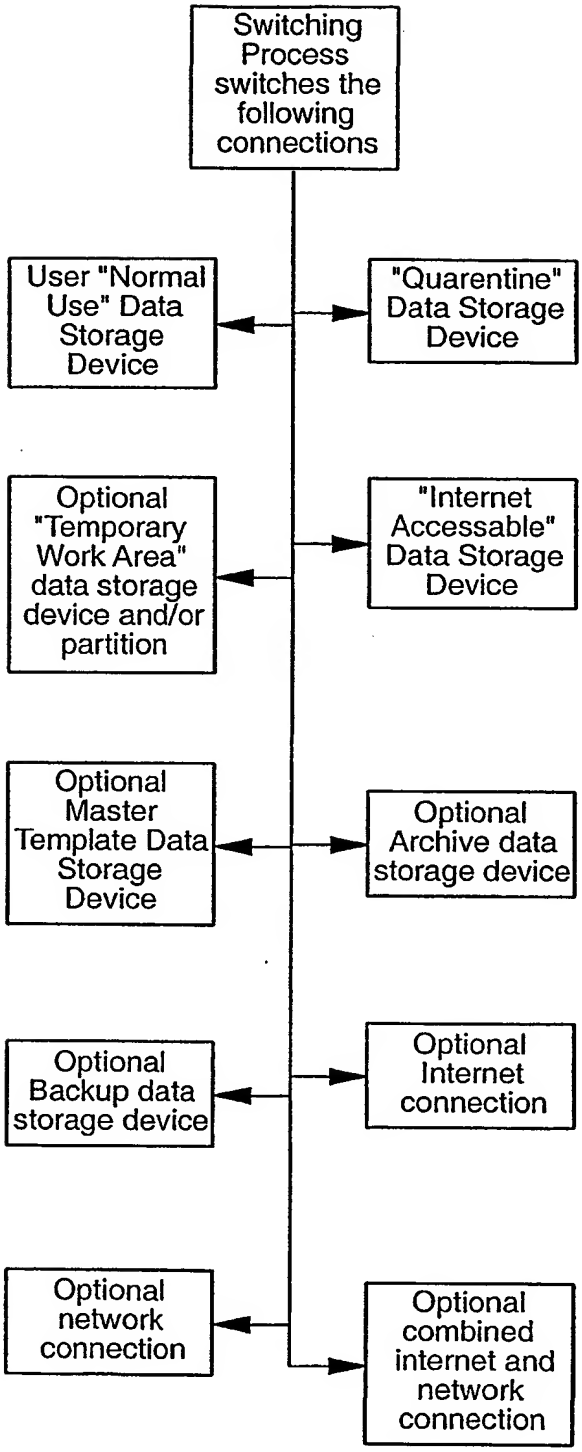

Switching Process switches the following connections
User "Normal Use" Data Storage Device
"Quarentine" Data Storage Device
Optional "Temporary Work Area" data storage device and/or partition
"Internet Accessable" Data Storage Device
Optional Master Template Data Storage Device
Optional Archive data storage device
Optional Backup data storage device
Optional Internet connection
Optional network connection
Optional combined internet and network connection

FIG._21Z-10

FIG._22A

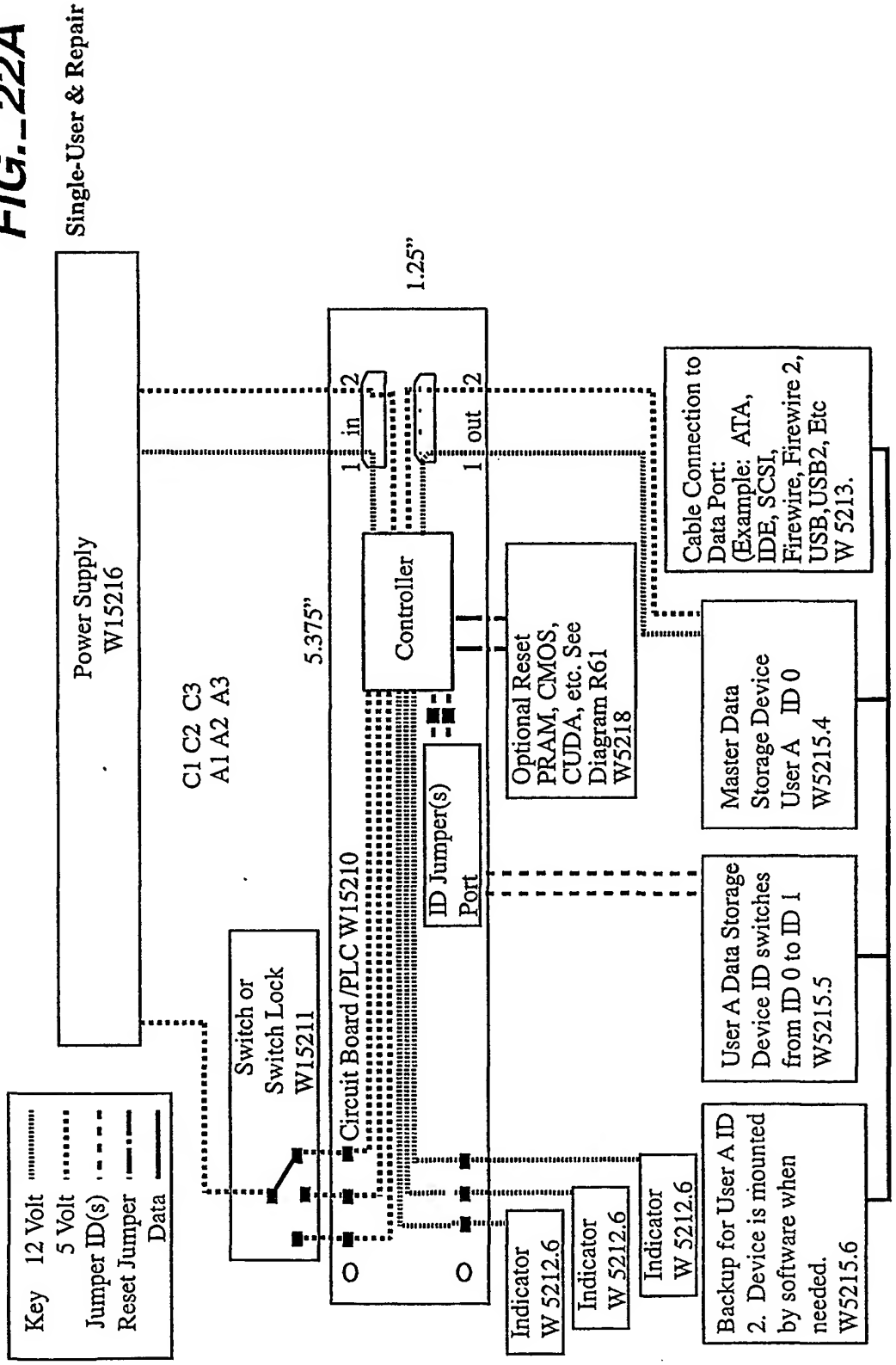

Single-User & Repair
Power Supply
W15216
C1 C2 C3
A1 A2 A3
5.375"
1.25"
Controller
1 in 2
1 out 2
Circuit Board /PLC W15210
ID Jumper(s)
Port
Switch or
Switch Lock
W15211
Optional Reset
PRAM, CMOS,
CUDA, etc. See
Diagram R61
W5218
Cable Connection to
Data Port:
(Example: ATA,
IDE, SCSI,
Firewire, Firewire 2,
USB,USB2, Etc
W 5213.
Master Data
Storage Device
User A ID 0
W5215.4
User A Data Storage
Device ID switches
from ID 0 to ID 1
W5215.5
Indicator
W 5212.6
Indicator
W 5212.6
Indicator
W 5212.6
Backup for User A ID
2. Device is mounted
by software when
needed.
W5215.6
Key 12 Volt
5 Volt
Jumper ID(s)
Reset Jumper
Data

FIG._22B

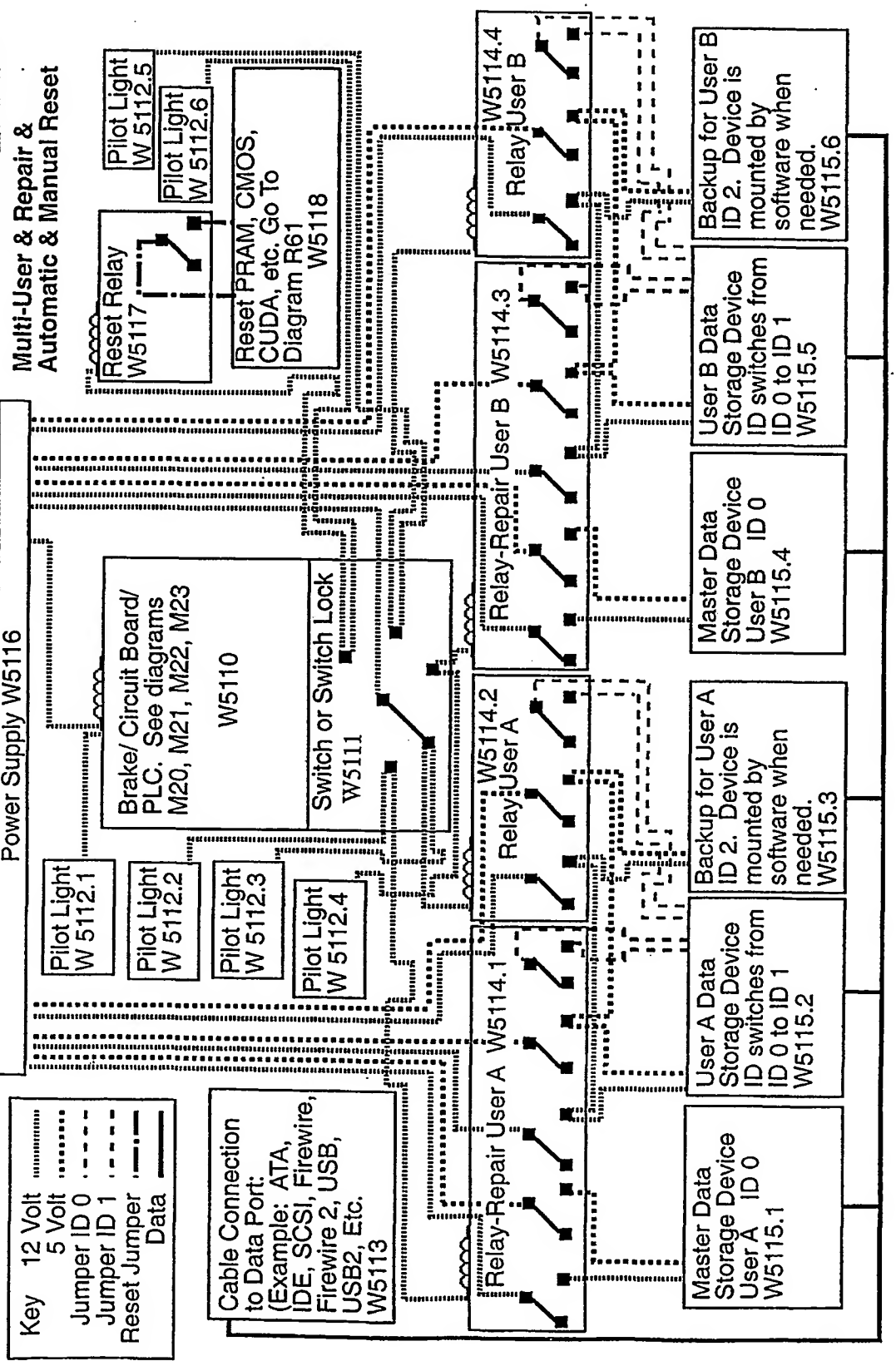

Multi-User & Repair &
Automatic & Manual Reset
Power Supply W5116
Reset Relay
W5117
Pilot Light
W 5112.5
Pilot Light
W 5112.6
Reset PRAM, CMOS,
CUDA, etc. Go To
Diagram R61
W5118
Brake/ Circuit Board/
PLC. See diagrams
M20, M21, M22, M23
W5110
Switch or Switch Lock
W5111
Pilot Light
W 5112.1
Pilot Light
W 5112.2
Pilot Light
W 5112.3
Pilot Light
W 5112.4
Relay-Repair User A W5114.1
Relay-Repair User A W5114.2
Relay-Repair User B W5114.3
Relay User B W5114.4
Master Data
Storage Device
User A ID 0
W5115.1
User A Data
Storage Device
ID switches from
ID 0 to ID 1
W5115.2
Backup for User A
ID 2. Device is
mounted by
software when
needed.
W5115.3
Master Data
Storage Device
User B ID 0
W5115.4
User B Data
Storage Device
ID switches from
ID 0 to ID 1
W5115.5
Backup for User B
ID 2. Device is
mounted by
software when
needed.
W5115.6
Cable Connection
to Data Port:
(Example: ATA,
IDE, SCSI, Firewire,
Firewire 2, USB,
USB2, Etc.
W5113
Key
12 Volt
5 Volt
Jumper ID 0
Jumper ID 1
Reset Jumper
Data

FIG._22C

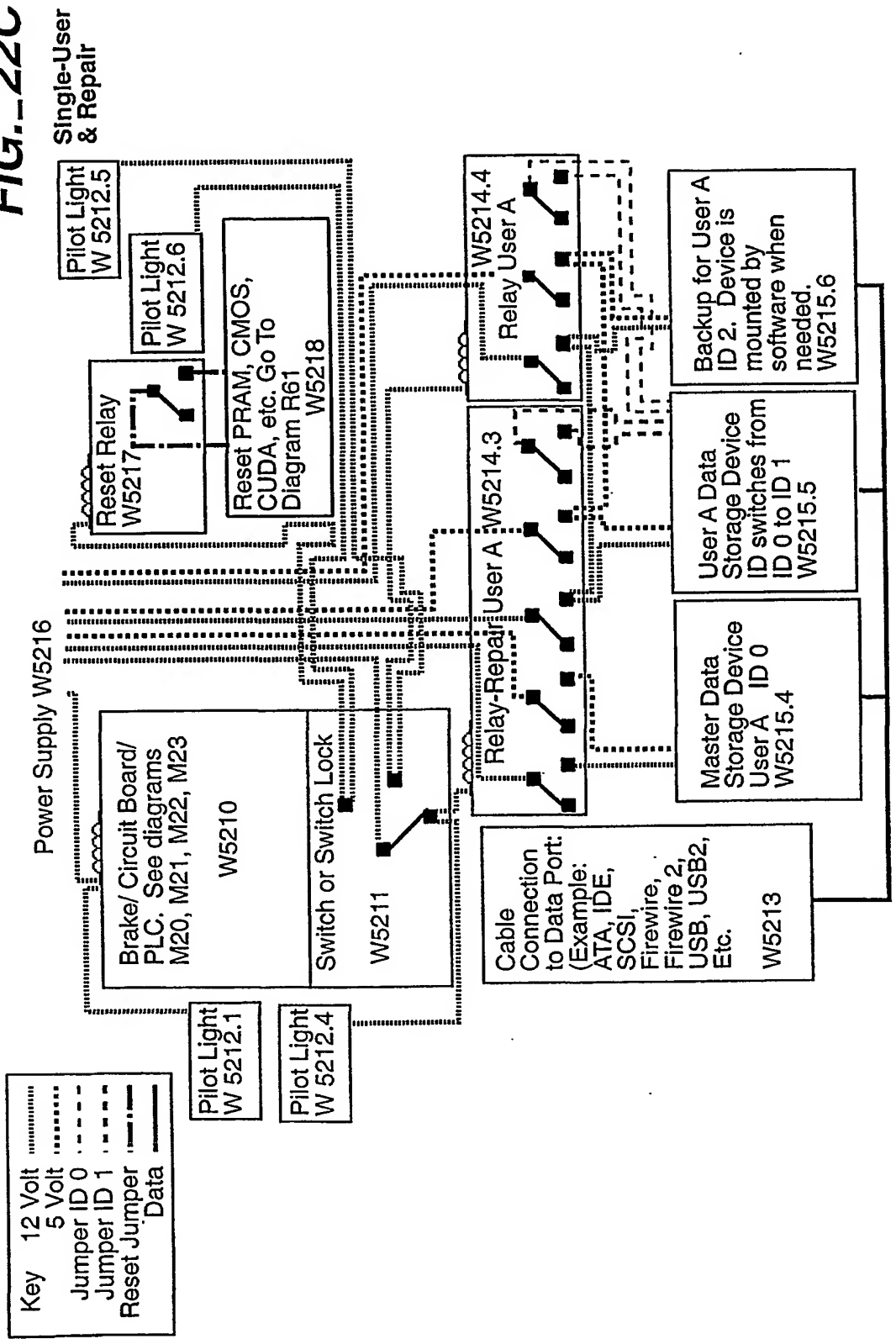
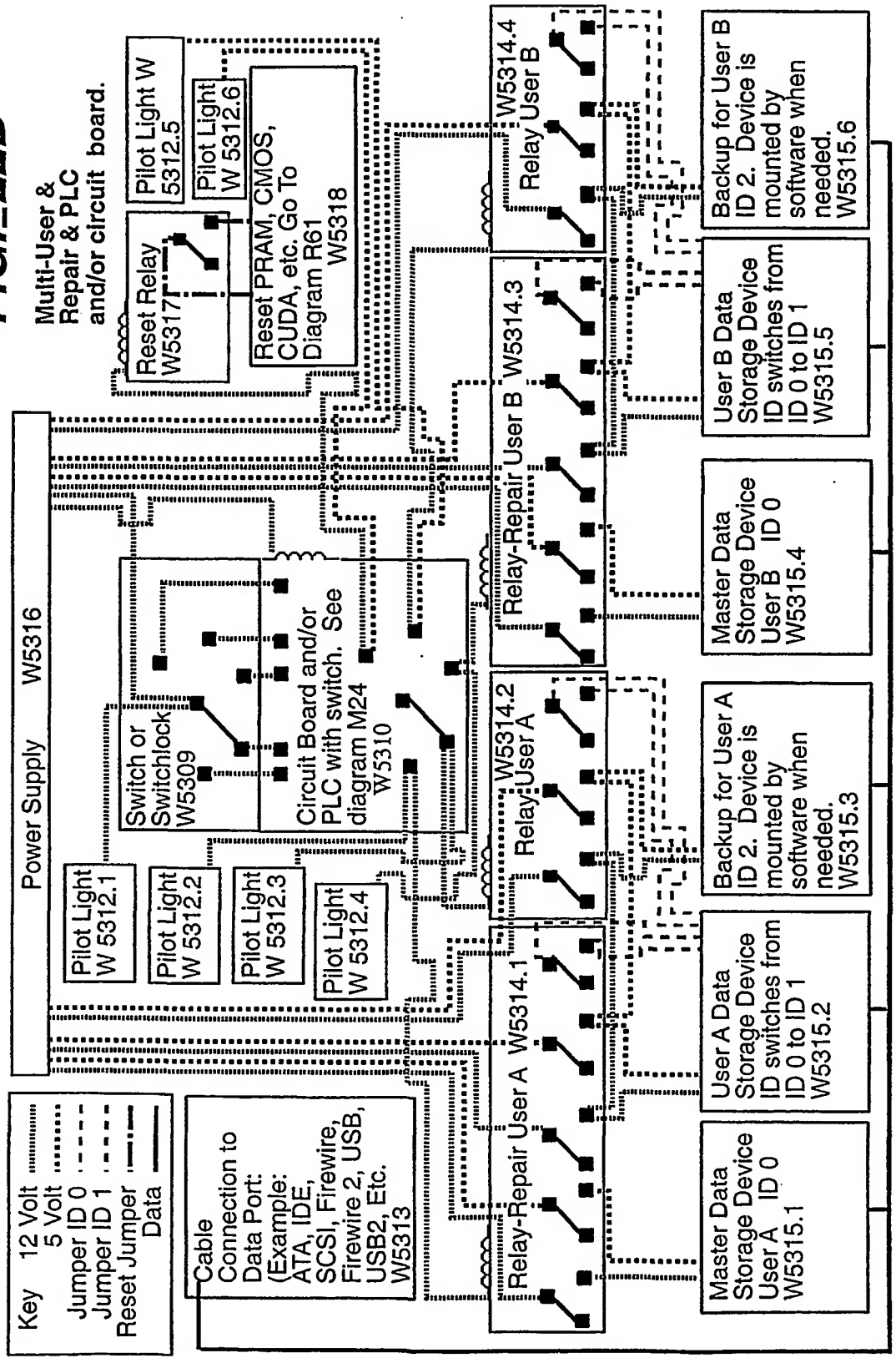

Single-User & Repair
Pilot Light W 5212.5
Pilot Light W 5212.6
Reset Relay W5217
Reset PRAM, CMOS, CUDA, etc. Go To Diagram R61 W5218
W5214.4 Relay User A
W5214.3 Relay-Repair User A
Backup for User A ID 2. Device is mounted by software when needed. W5215.6
User A Data Storage Device ID switches from ID 0 to ID 1 W5215.5
Master Data Storage Device User A ID 0 W5215.4
Cable Connection to Data Port: (Example: ATA, IDE, SCSI, Firewire, Firewire 2, USB, USB2, Etc. W5213
Power Supply W5216
Brake/ Circuit Board/ PLC. See diagrams M20, M21, M22, M23 W5210
Switch or Switch Lock W5211
Pilot Light W 5212.1
Pilot Light W 5212.4
Key
12 Volt
5 Volt
Jumper ID 0
Jumper ID 1
Reset Jumper
Data

FIG._22D


Multi-User & Repair & PLC and/or circuit board.
Key
12 Volt
5 Volt
Jumper ID 0
Jumper ID 1
Reset Jumper
Data
Power Supply W5316
Cable Connection to Data Port: (Example: ATA, IDE, SCSI, Firewire, Firewire 2, USB, USB2, Etc. W5313
Pilot Light W 5312.1
Pilot Light W 5312.2
Pilot Light W 5312.3
Pilot Light W 5312.4
Switch or Switchlock W5309
Circuit Board and/or PLC with switch. See diagram M24 W5310
Reset Relay W5317
Pilot Light W 5312.5
Pilot Light W 5312.6
Reset PRAM, CMOS, CUDA, etc. Go To Diagram R61 W5318
Relay-Repair User A W5314.1
Relay User A W5314.2
Relay-Repair User B W5314.3
Relay User B W5314.4
Master Data Storage Device User A ID 0 W5315.1
User A Data Storage Device ID switches from ID 0 to ID 1 W5315.2
Backup for User A ID 2. Device is mounted by software when needed. W5315.3
Master Data Storage Device User B ID 0 W5315.4
User B Data Storage Device ID switches from ID 0 to ID 1 W5315.5
Backup for User B ID 2. Device is mounted by software when needed. W5315.6

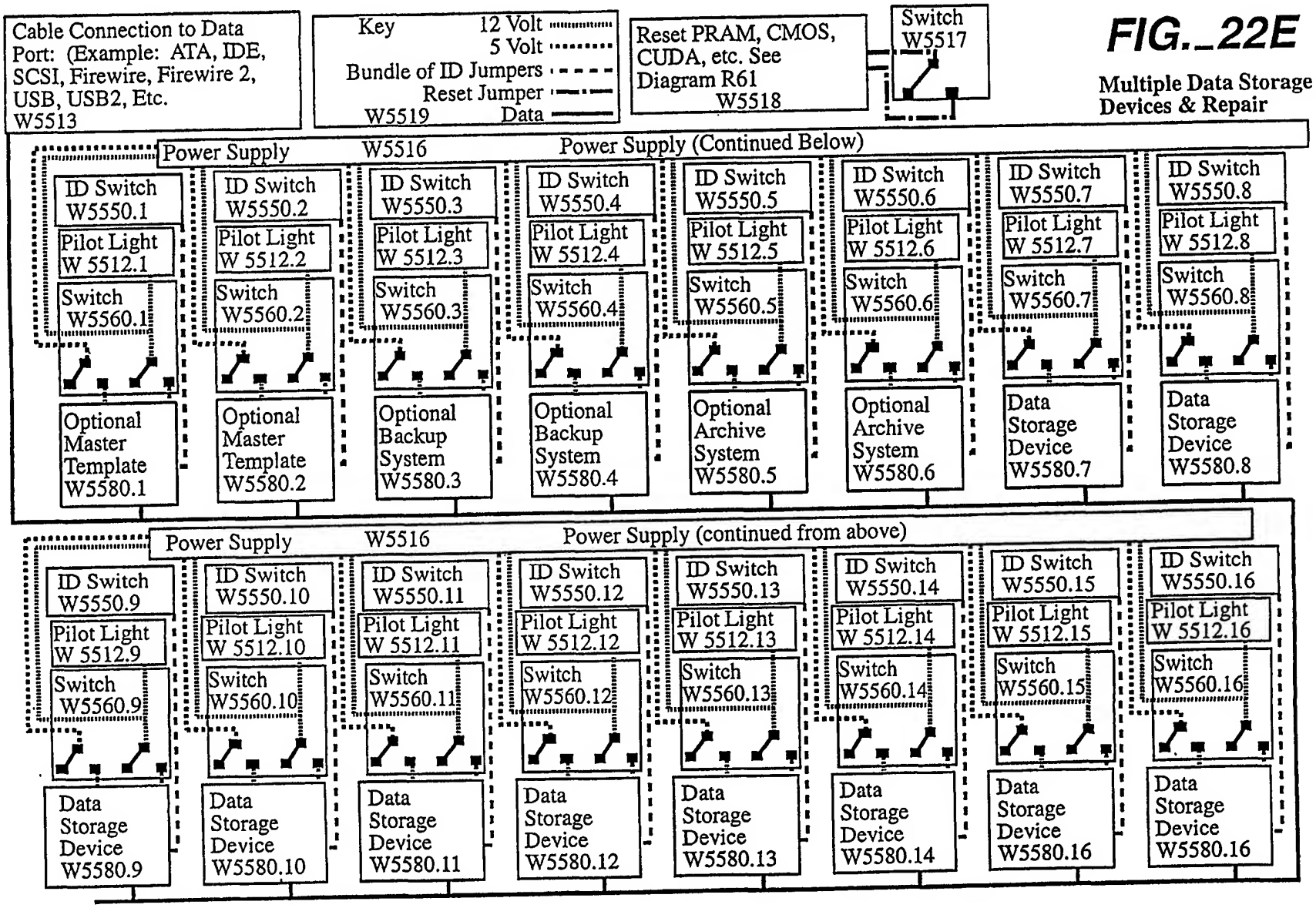

Cable Connection to Data Port: (Example: ATA, IDE, SCSI, Firewire, Firewire 2, USB, USB2, Etc.
W5513
Key
12 Volt
5 Volt
Bundle of ID Jumpers
Reset Jumper
Data
W5519
Reset PRAM, CMOS, CUDA, etc. See Diagram R61
W5518
Switch W5517
Multiple Data Storage Devices & Repair
Power Supply W5516
Power Supply (Continued Below)
ID Switch W5550.1
Pilot Light W 5512.1
Switch W5560.1
Optional Master Template W5580.1
ID Switch W5550.2
Pilot Light W 5512.2
Switch W5560.2
Optional Master Template W5580.2
ID Switch W5550.3
Pilot Light W 5512.3
Switch W5560.3
Optional Backup System W5580.3
ID Switch W5550.4
Pilot Light W 5512.4
Switch W5560.4
Optional Backup System W5580.4
ID Switch W5550.5
Pilot Light W 5512.5
Switch W5560.5
Optional Archive System W5580.5
ID Switch W5550.6
Pilot Light W 5512.6
Switch W5560.6
Optional Archive System W5580.6
ID Switch W5550.7
Pilot Light W 5512.7
Switch W5560.7
Data Storage Device W5580.7
ID Switch W5550.8
Pilot Light W 5512.8
Switch W5560.8
Data Storage Device W5580.8
Power Supply W5516
Power Supply (continued from above)
ID Switch W5550.9
Pilot Light W 5512.9
Switch W5560.9
Data Storage Device W5580.9
ID Switch W5550.10
Pilot Light W 5512.10
Switch W5560.10
Data Storage Device W5580.10
ID Switch W5550.11
Pilot Light W 5512.11
Switch W5560.11
Data Storage Device W5580.11
ID Switch W5550.12
Pilot Light W 5512.12
Switch W5560.12
Data Storage Device W5580.12
ID Switch W5550.13
Pilot Light W 5512.13
Switch W5560.13
Data Storage Device W5580.13
ID Switch W5550.14
Pilot Light W 5512.14
Switch W5560.14
Data Storage Device W5580.14
ID Switch W5550.15
Pilot Light W 5512.15
Switch W5560.15
Data Storage Device W5580.16
ID Switch W5550.16
Pilot Light W 5512.16
Switch W5560.16
Data Storage Device W5580.16

FIG._22E

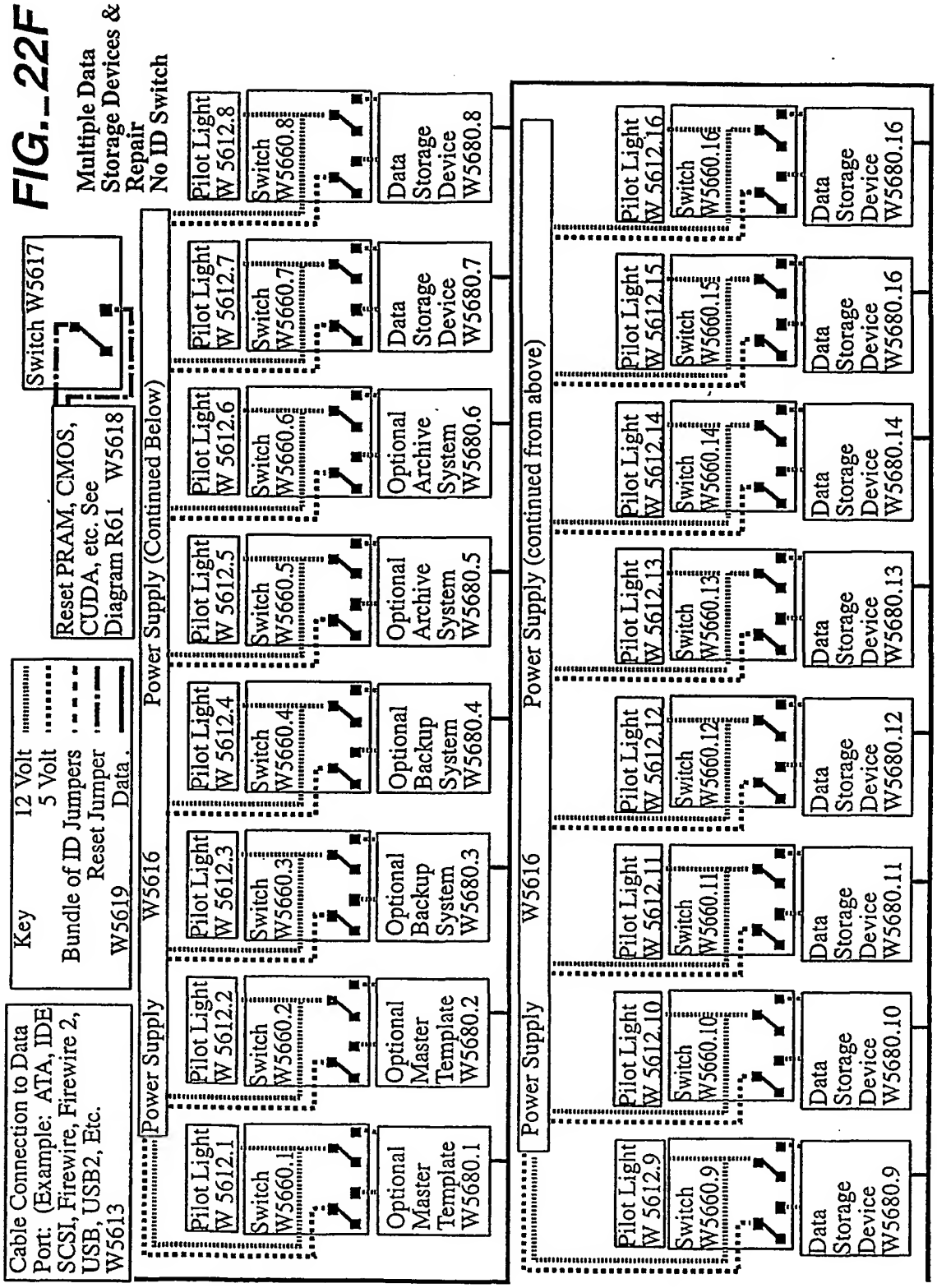

Cable Connection to Data Port: (Example: ATA, IDE SCSI, Firewire, Firewire 2, USB, USB2, Etc. W5613
Key 12 Volt 5 Volt
Bundle of ID Jumpers
Reset Jumper
Data
W5619
Reset PRAM, CMOS, CUDA, etc. See Diagram R61 W5618
Switch W5617
Multiple Data Storage Devices & Repair No ID Switch
Power Supply W5616 Power Supply (Continued Below)
Power Supply W5616 Power Supply (continued from above)
Pilot Light W 5612.1
Switch W5660.1
Optional Master Template W5680.1
Pilot Light W 5612.2
Switch W5660.2
Optional Master Template W5680.2
Pilot Light W 5612.3
Switch W5660.3
Optional Backup System W5680.3
Pilot Light W 5612.4
Switch W5660.4
Optional Backup System W5680.4
Pilot Light W 5612.5
Switch W5660.5
Optional Archive System W5680.5
Pilot Light W 5612.6
Switch W5660.6
Optional Archive System W5680.6
Pilot Light W 5612.7
Switch W5660.7
Data Storage Device W5680.7
Pilot Light W 5612.8
Switch W5660.8
Data Storage Device W5680.8
Pilot Light W 5612.9
Switch W5660.9
Data Storage Device W5680.9
Pilot Light W 5612.10
Switch W5660.10
Data Storage Device W5680.10
Pilot Light W 5612.11
Switch W5660.11
Data Storage Device W5680.11
Pilot Light W 5612.12
Switch W5660.12
Data Storage Device W5680.12
Pilot Light W 5612.13
Switch W5660.13
Data Storage Device W5680.13
Pilot Light W 5612.14
Switch W5660.14
Data Storage Device W5680.14
Pilot Light W 5612.15
Switch W5660.15
Data Storage Device W5680.16
Pilot Light W 5612.16
Switch W5660.16
Data Storage Device W5680.16

FIG._22F

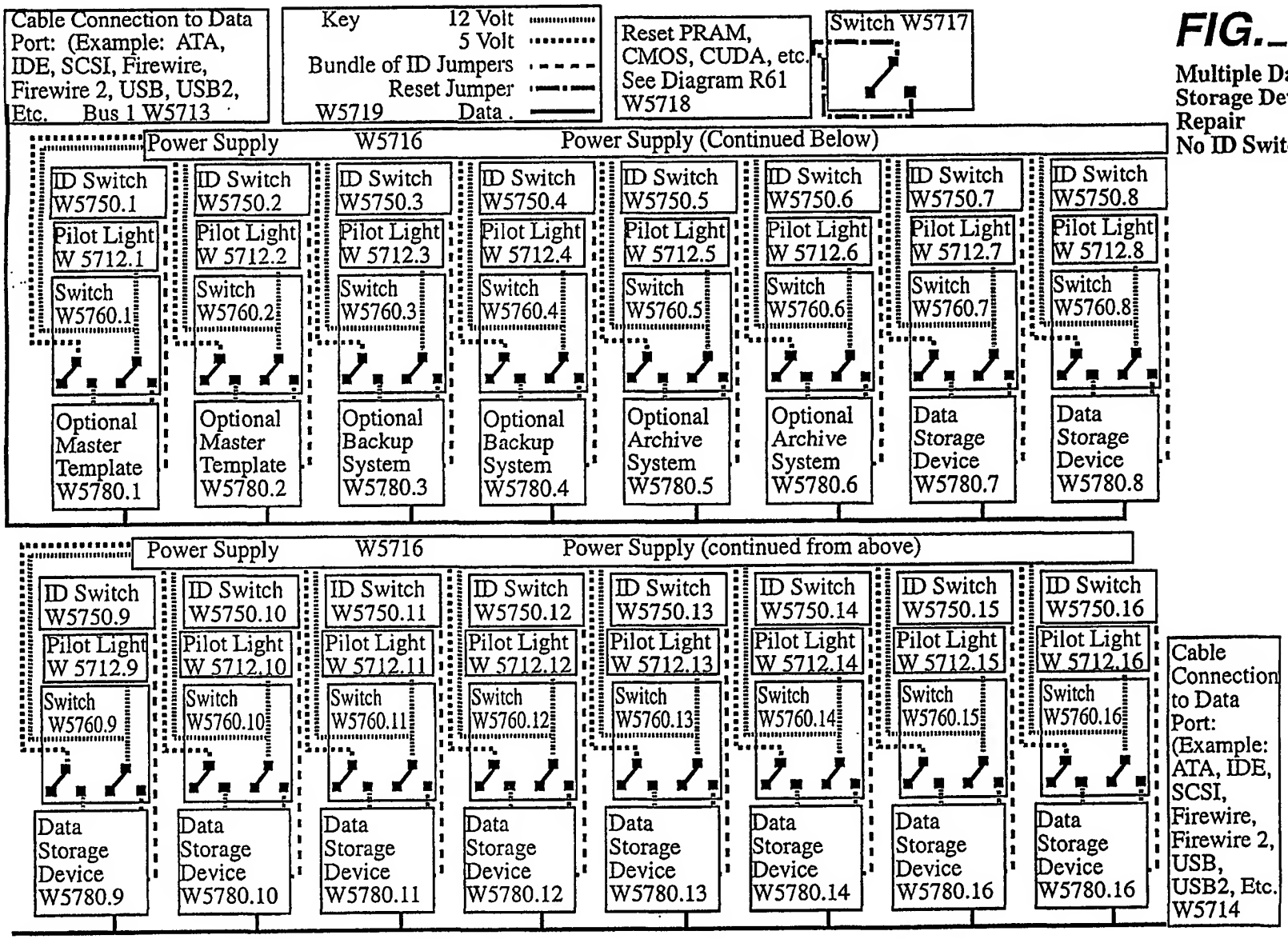


FIG._22G

Multiple Data Storage Devices & Repair No ID Switch

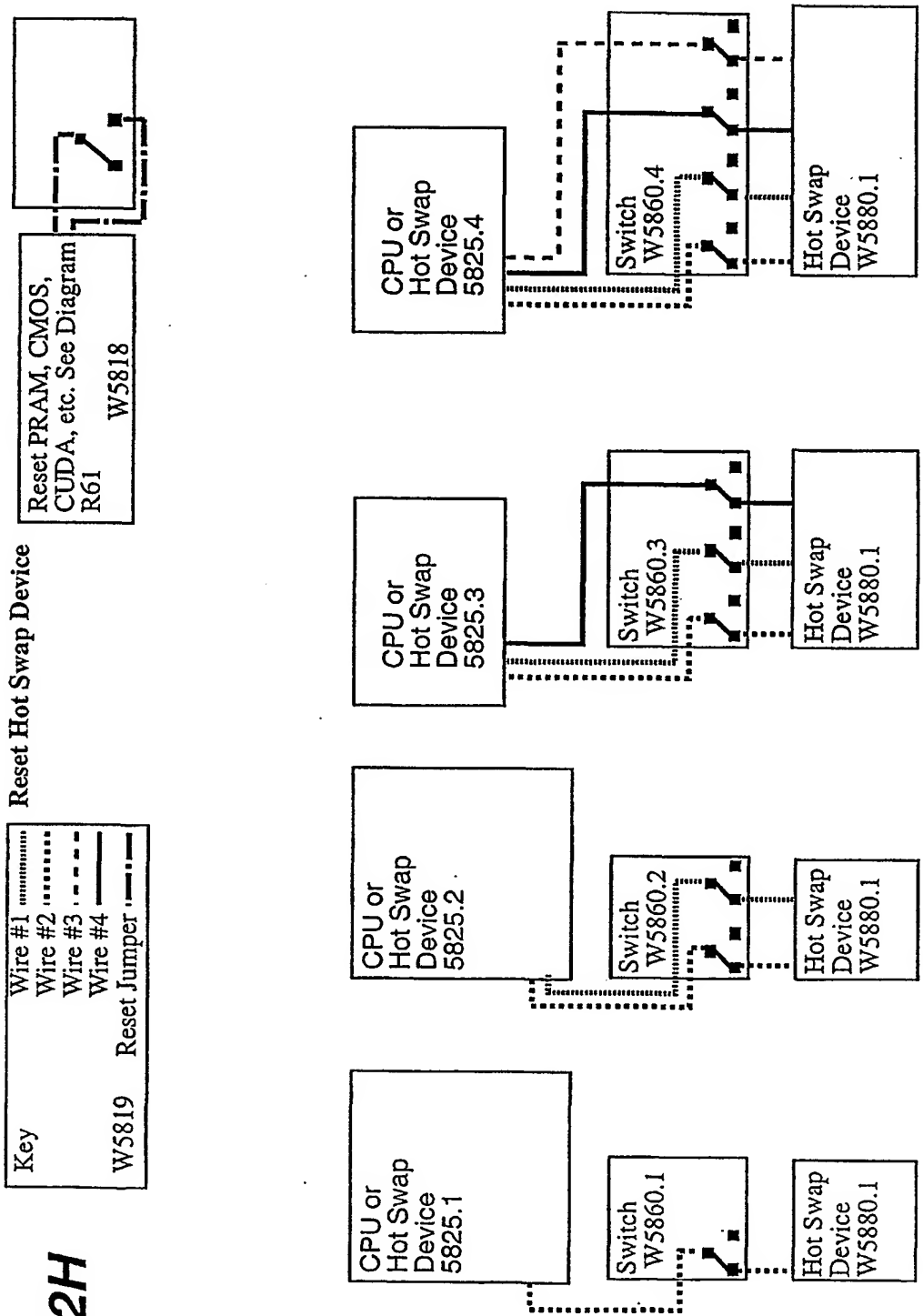

Reset Hot Swap Device
Key
Wire #1
Wire #2
Wire #3
Wire #4
W5819
Reset Jumper
Reset PRAM, CMOS, CUDA, etc. See Diagram R61
W5818
CPU or Hot Swap Device 5825.1
Switch W5860.1
Hot Swap Device W5880.1
CPU or Hot Swap Device 5825.2
Switch W5860.2
Hot Swap Device W5880.1
CPU or Hot Swap Device 5825.3
Switch W5860.3
Hot Swap Device W5880.1
CPU or Hot Swap Device 5825.4
Switch W5860.4
Hot Swap Device W5880.1

FIG._22H

FIG._22I

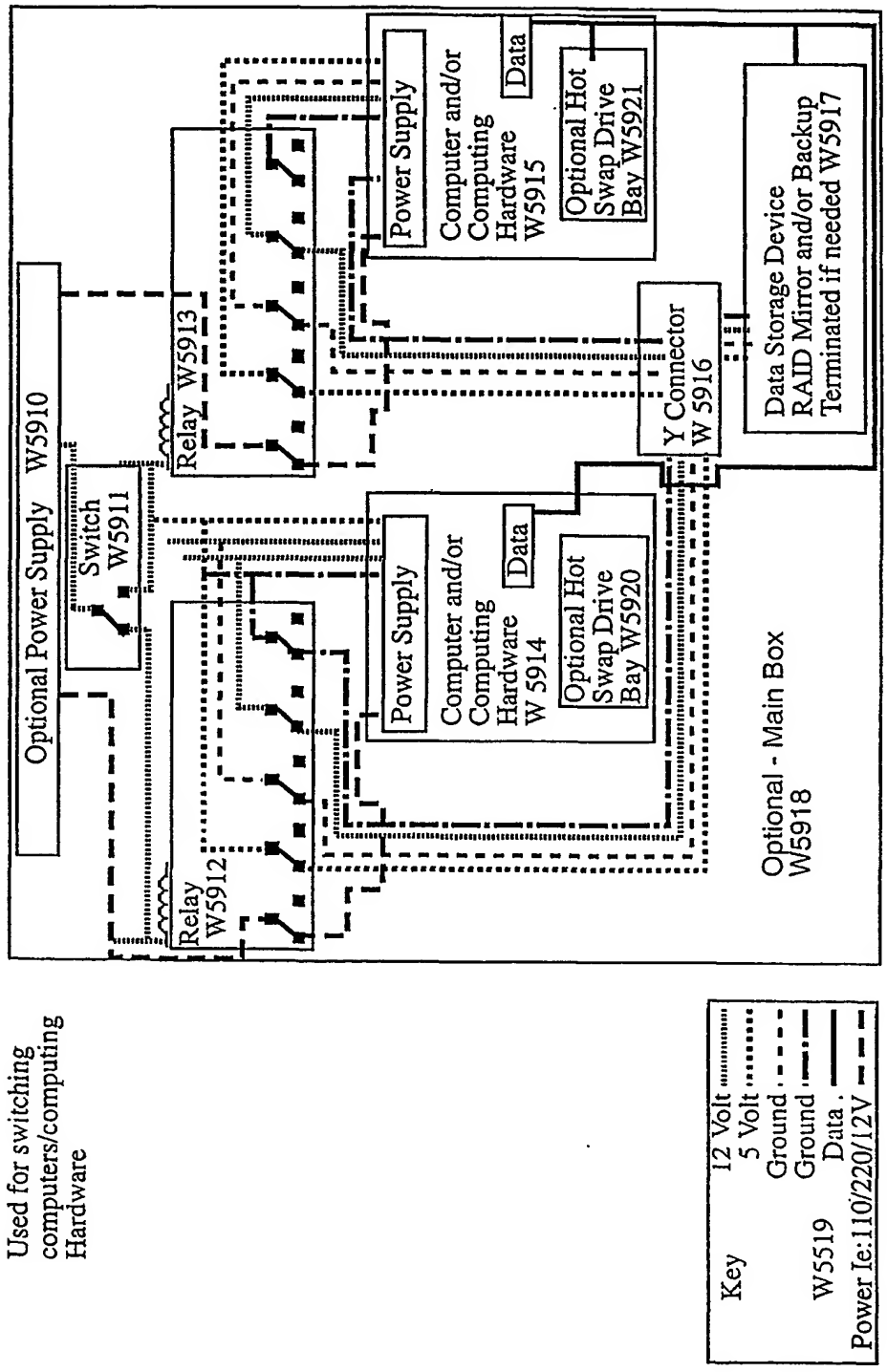

Optional Power Supply W5910
Switch W5911
Relay W5912
Relay W5913
Power Supply
Computer and/or Computing Hardware W5915
Data
Optional Hot Swap Drive Bay W5921
Power Supply
Computer and/or Computing Hardware W 5914
Data
Optional Hot Swap Drive Bay W5920
Y Connector W 5916
Data Storage Device RAID Mirror and/or Backup Terminated if needed W5917
Optional - Main Box W5918
Used for switching computers/computing Hardware
Key
12 Volt
5 Volt
Ground
Ground
W5519 Data
Power Ie:110/220/12V

FIG._22J

Switching between computers and/or computing hardware.

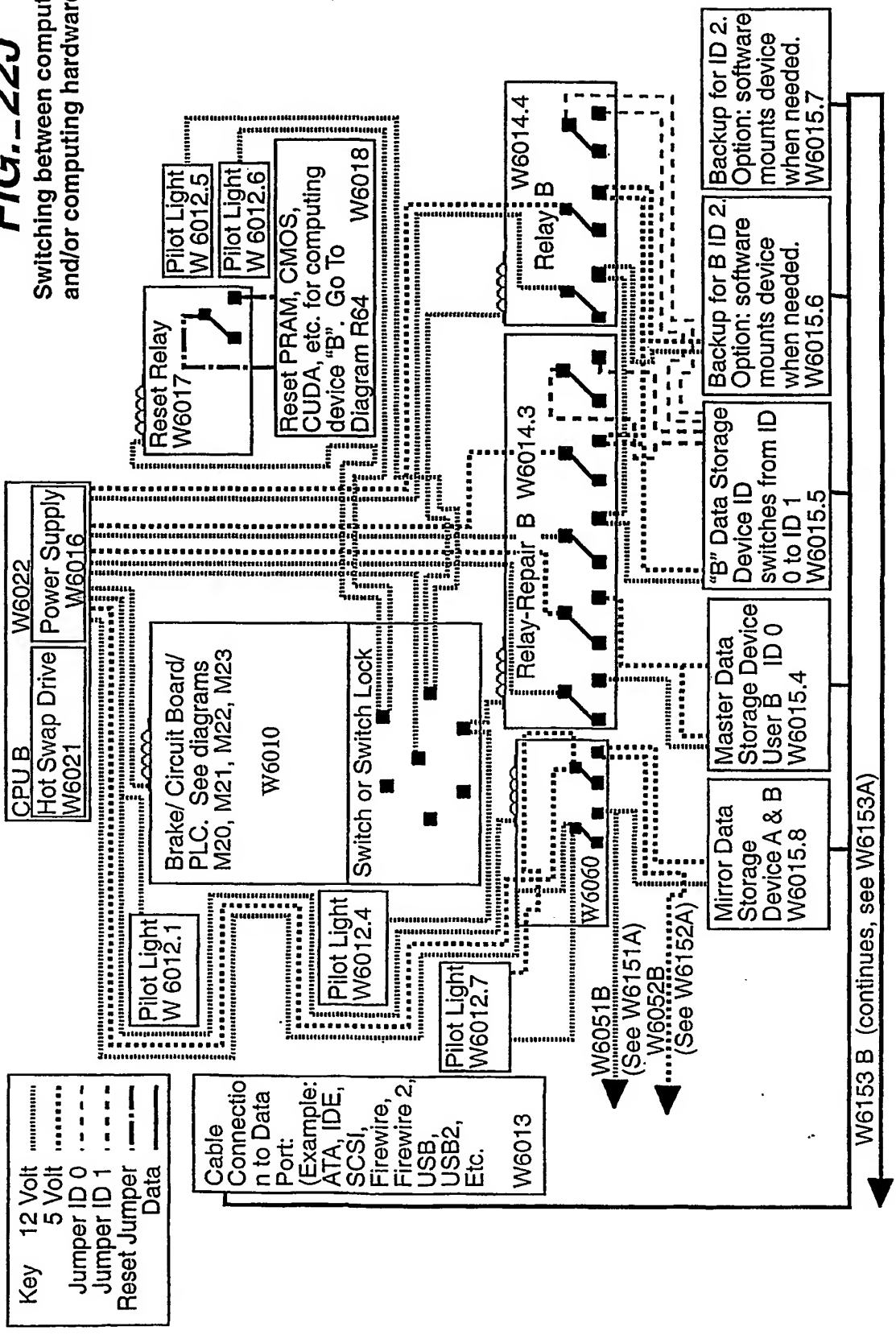


FIG._22K

Switching between computers and/or computing hardware.

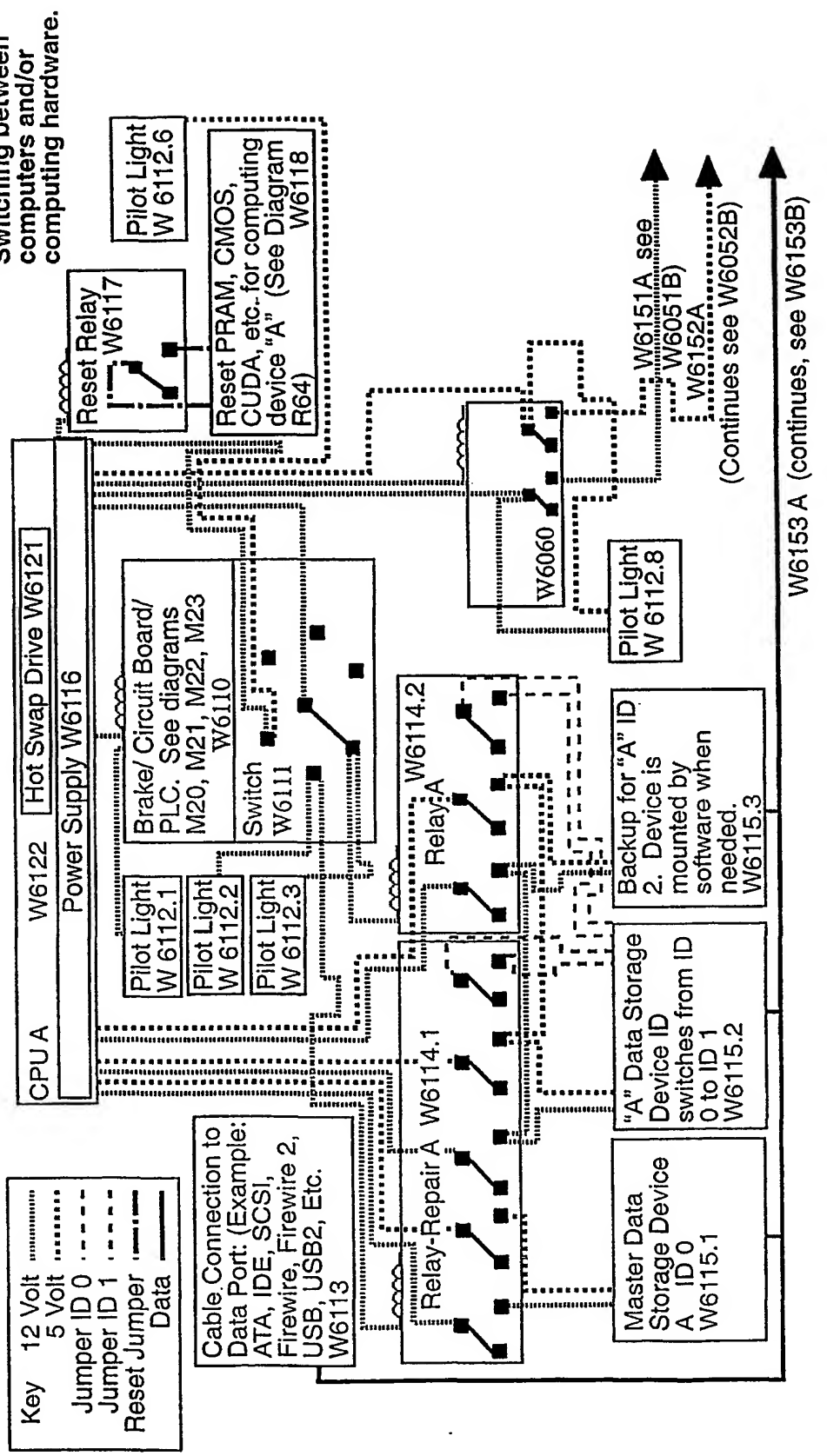


FIG._22L
used for switching computers and/or computing Hardware

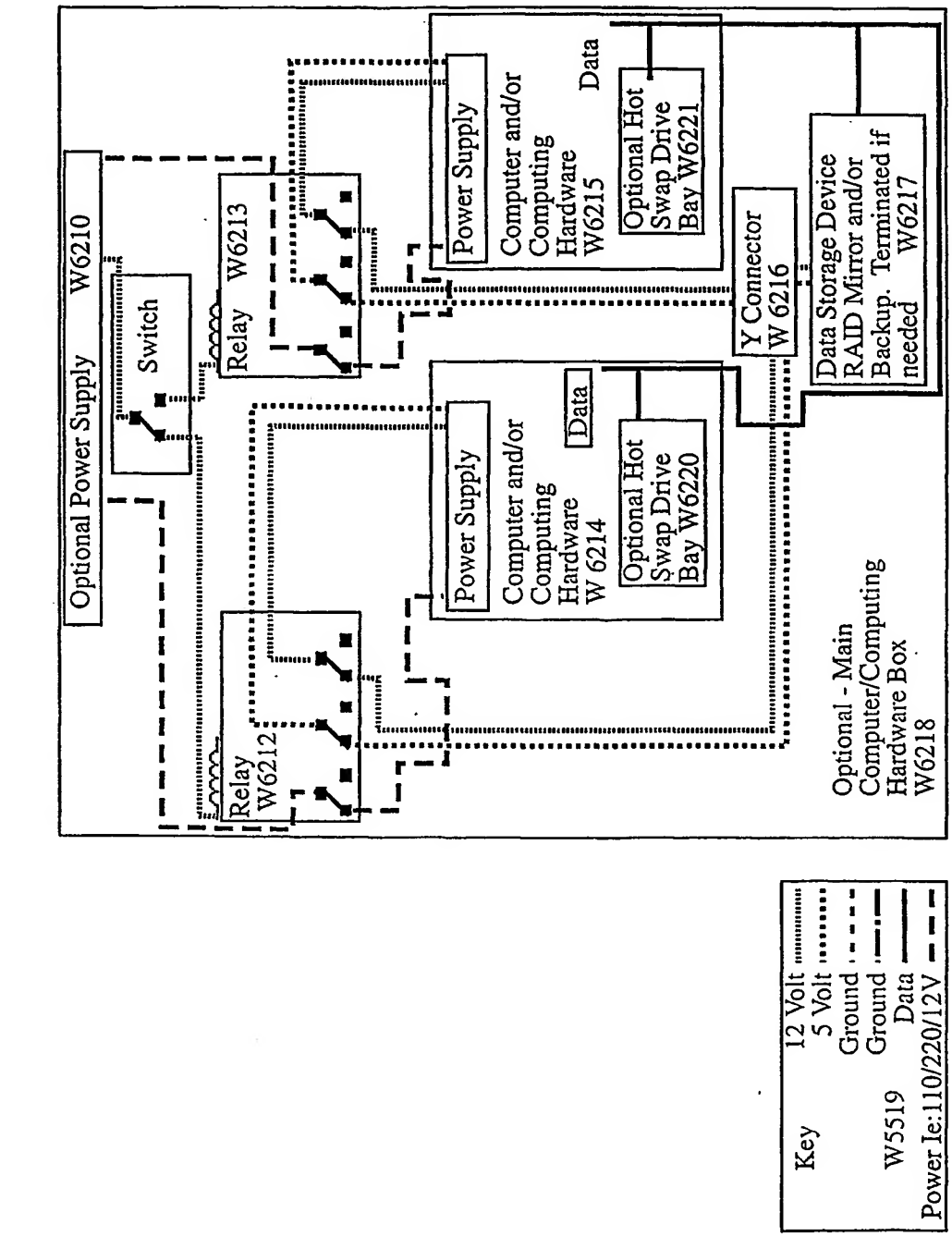
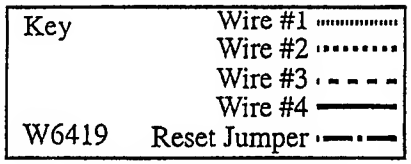
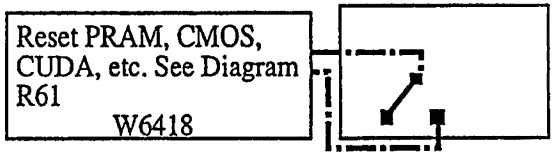


FIG._22M



DRAMUS
Switch connection to global computer/communication network, and other networks.



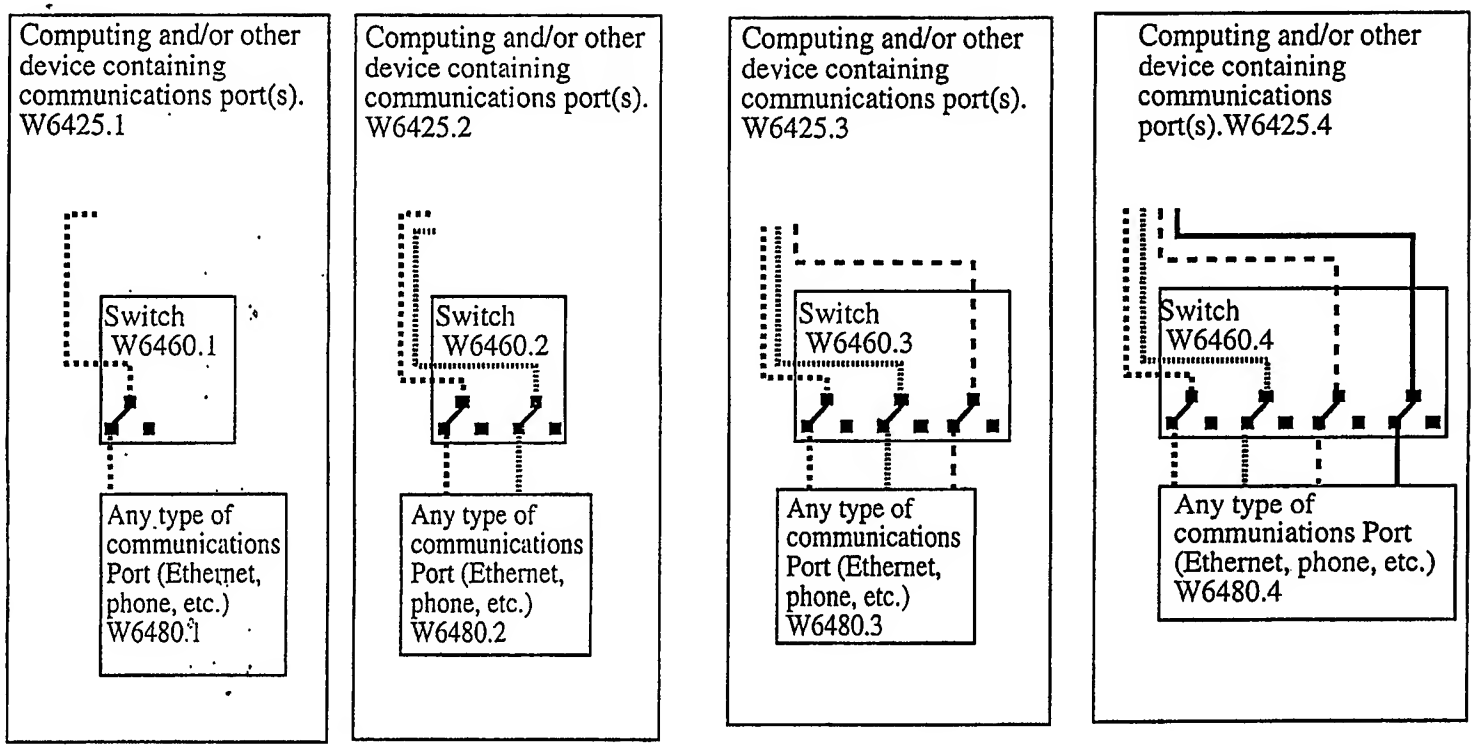
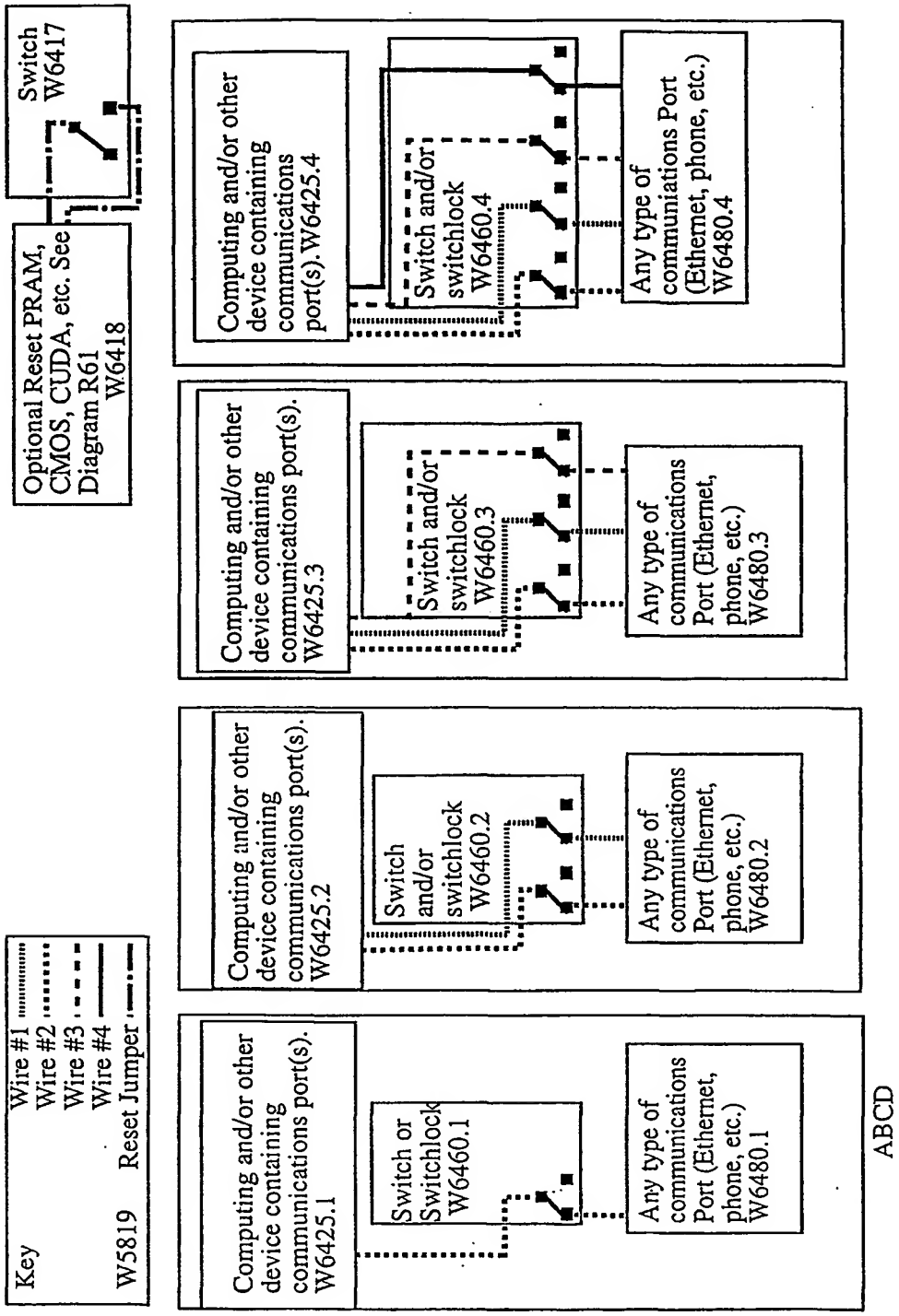


FIG._22N


Key
Wire #1
Wire #2
Wire #3
Wire #4
W5819 Reset Jumper
Optional Reset PRAM, CMOS, CUDA, etc. See Diagram R61 W6418
Switch W6417
Computing and/or other device containing communications port(s). W6425.1
Switch or Switchlock W6460.1
Any type of communications Port (Ethernet, phone, etc.) W6480.1
Computing and/or other device containing communications port(s). W6425.2
Switch and/or switchlock W6460.2
Any type of communications Port (Ethernet, phone, etc.) W6480.2
Computing and/or other device containing communications port(s). W6425.3
Switch and/or switchlock W6460.3
Any type of communications Port (Ethernet, phone, etc.) W6480.3
Computing and/or other device containing communications port(s). W6425.4
Switch and/or switchlock W6460.4
Any type of communiations Port (Ethernet, phone, etc.) W6480.4
ABCD

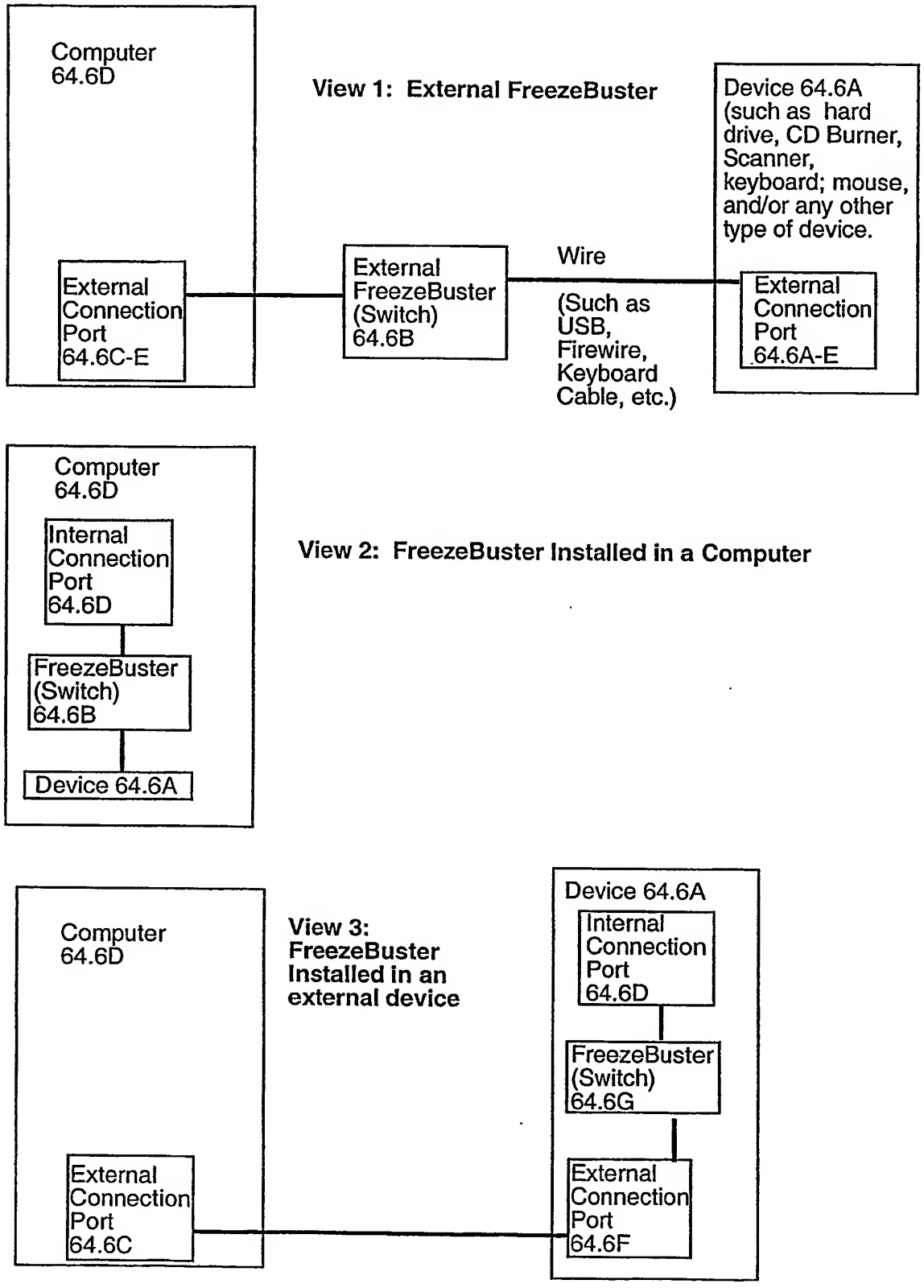

Computer
64.6D
External
Connection
Port
64.6C-E
View 1: External FreezeBuster
External
FreezeBuster
(Switch)
64.6B
Wire
(Such as
USB,
Firewire,
Keyboard
Cable, etc.)
Device 64.6A
(such as hard
drive, CD Burner,
Scanner,
keyboard; mouse,
and/or any other
type of device.
External
Connection
Port
64.6A-E
Computer
64.6D
Internal
Connection
Port
64.6D
FreezeBuster
(Switch)
64.6B
Device 64.6A
View 2: FreezeBuster Installed in a Computer
Computer
64.6D
External
Connection
Port
64.6C
View 3:
FreezeBuster
Installed in an
external device
Device 64.6A
Internal
Connection
Port
64.6D
FreezeBuster
(Switch)
64.6G
External
Connection
Port
64.6F

FIG._220

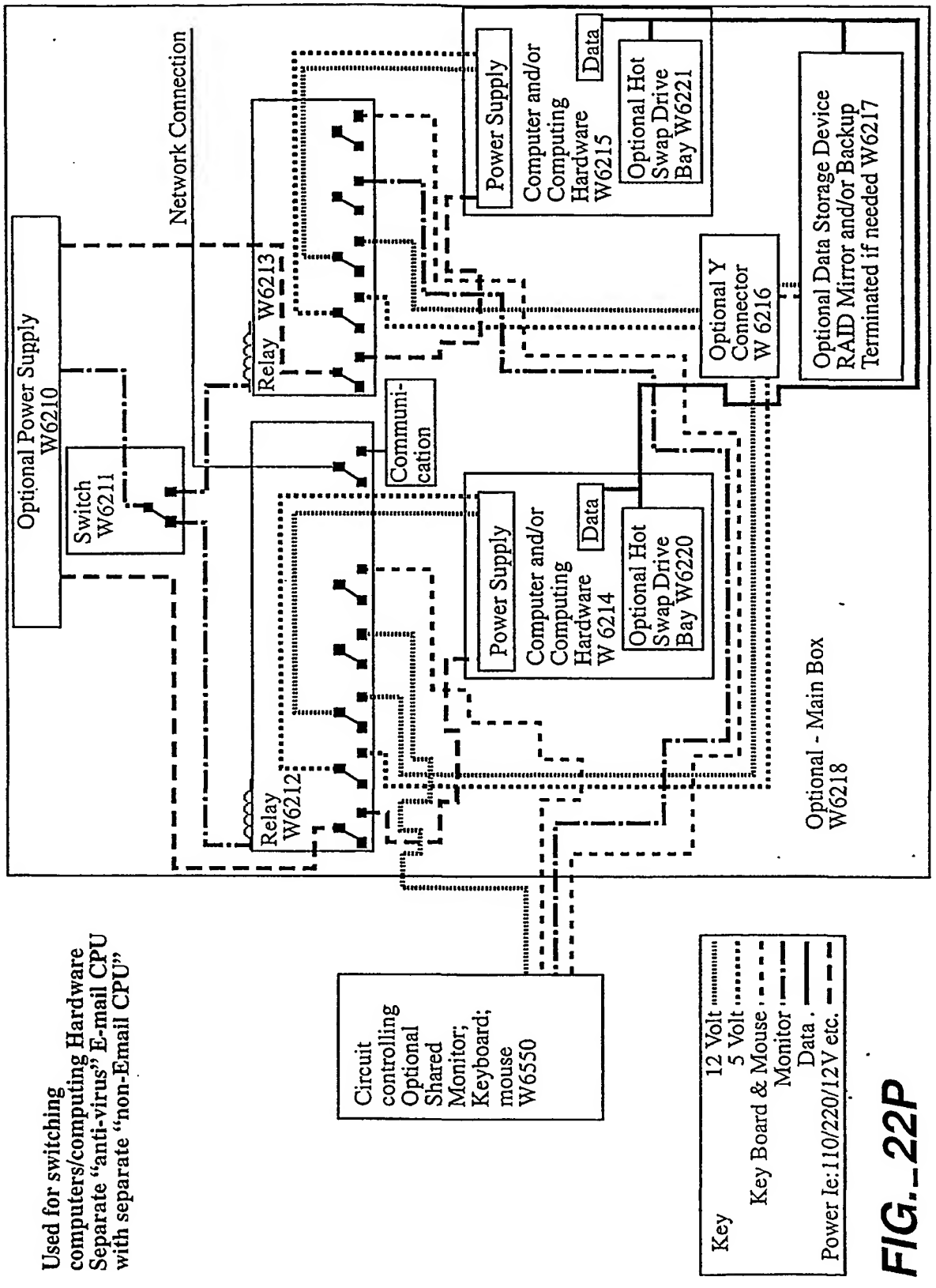
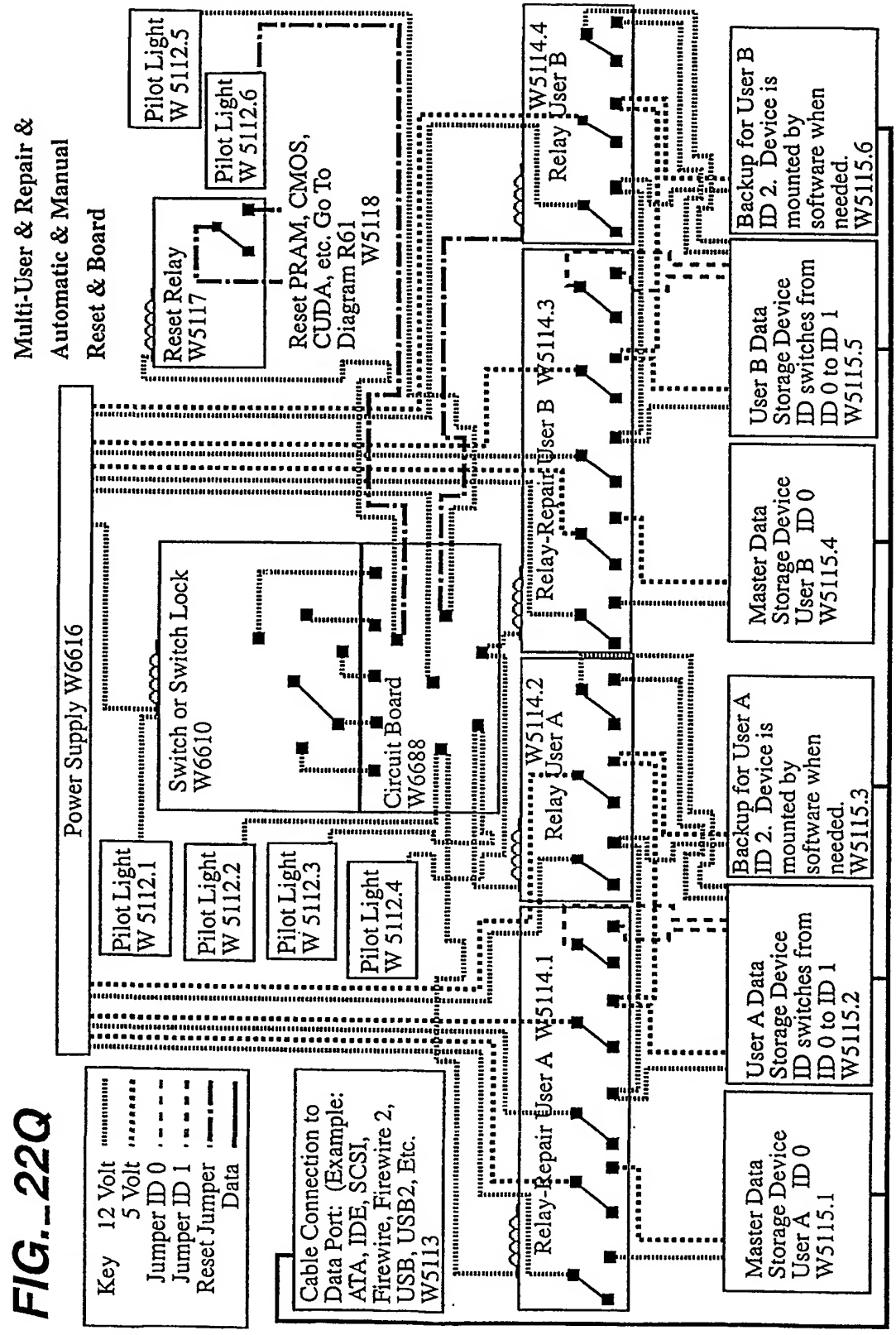

Optional Power Supply W6210
Switch W6211
Network Connection
Relay W6213
Relay W6212
Communi-cation
Power Supply
Computer and/or Computing Hardware W6215
Data
Optional Hot Swap Drive Bay W6221
Optional Y Connector W 6216
Optional Data Storage Device RAID Mirror and/or Backup Terminated if needed W6217
Power Supply
Computer and/or Computing Hardware W 6214
Data
Optional Hot Swap Drive Bay W6220
Optional - Main Box W6218
Circuit controlling Optional Shared Monitor; Keyboard; mouse W6550
Key
12 Volt
5 Volt
Key Board & Mouse
Monitor
Data
Power Ie:110/220/12V etc.
Used for switching computers/computing Hardware Separate "anti-virus" E-mail CPU with separate "non-Email CPU"

FIG._22P

FIG._22Q


Multi-User & Repair &
Automatic & Manual
Reset & Board
Power Supply W6616
Key
12 Volt
5 Volt
Jumper ID 0
Jumper ID 1
Reset Jumper
Data
Cable Connection to Data Port: (Example: ATA, IDE, SCSI, Firewire, Firewire 2, USB, USB2, Etc. W5113
Pilot Light W 5112.1
Pilot Light W 5112.2
Pilot Light W 5112.3
Pilot Light W 5112.4
Switch or Switch Lock W6610
Circuit Board W6688
Reset Relay W5117
Reset PRAM, CMOS, CUDA, etc. Go To Diagram R61 W5118
Pilot Light W 5112.6
Pilot Light W 5112.5
Relay-Repair User A W5114.1
Relay User A W5114.2
Relay-Repair User B W5114.3
Relay User B W5114.4
Master Data Storage Device User A ID 0 W5115.1
User A Data Storage Device ID switches from ID 0 to ID 1 W5115.2
Backup for User A ID 2. Device is mounted by software when needed. W5115.3
Master Data Storage Device User B ID 0 W5115.4
User B Data Storage Device ID switches from ID 0 to ID 1 W5115.5
Backup for User B ID 2. Device is mounted by software when needed. W5115.6

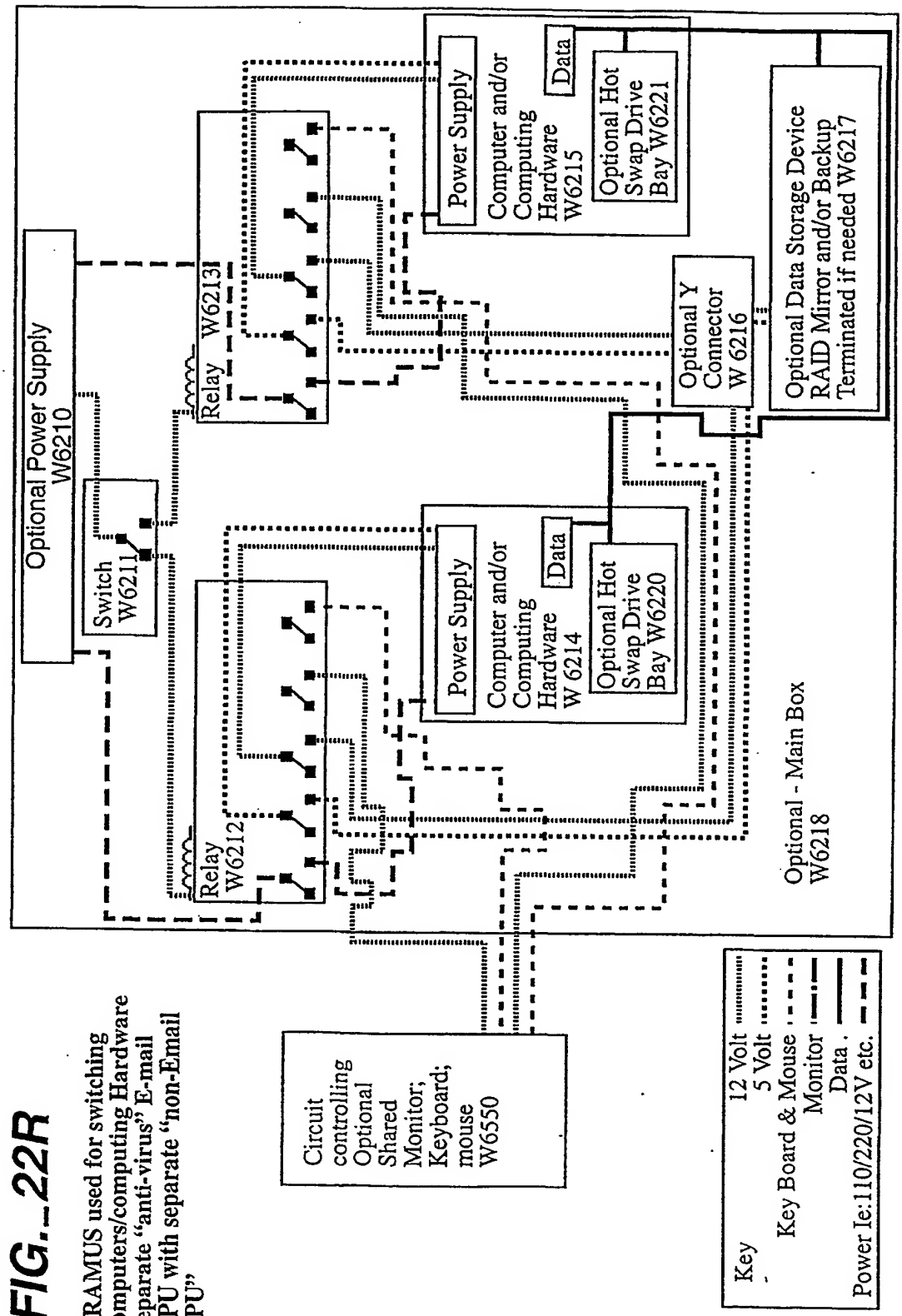
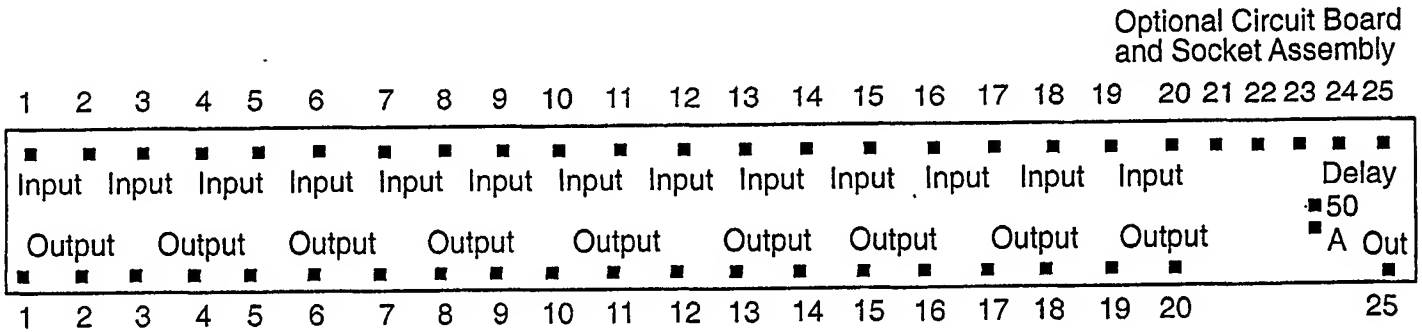


FIG._22R

DRAMUS used for switching computers/computing Hardware Separate "anti-virus" E-mail CPU with separate "non-Email CPU"



1) circuit 1
2) circuit 2
3) circuit 3
4) circuit 4
5) circuit 5
6) circuit 6
7) circuit 7
8) circuit 8
9) circuit 9
10) circuit 10
11) circuit 11
12) circuit 12
13) circuit 13
14) circuit 14
15) circuit 15
16) circuit 16
17) circuit 17
18) circuit 18
19) circuit 19
20) circuit 20
21) Power Control Indicator #21
22) Power Control Indicator #22
23) Power Control Indicator #23
24) Power Control Indicator #24
25) Power Control Indicator #25
26) time delay circuit
27) Data and power to LCD screen and/or data for computer monitor and/or to computer.
28) Power to board
40) delay start time for delay circuit
41) stop time for delay circuit
50) jumper "A"

FIG._22S

FIG._22T

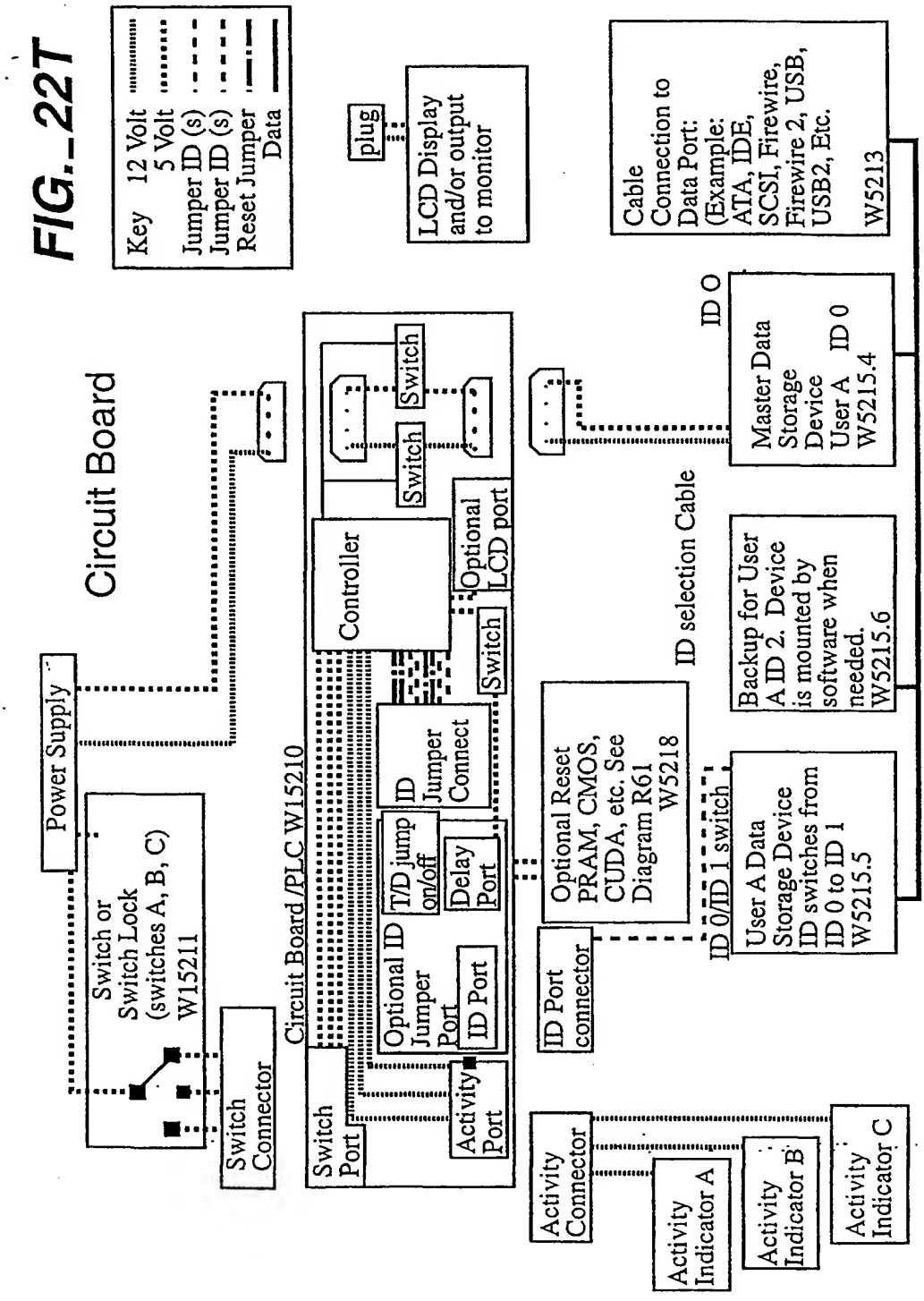
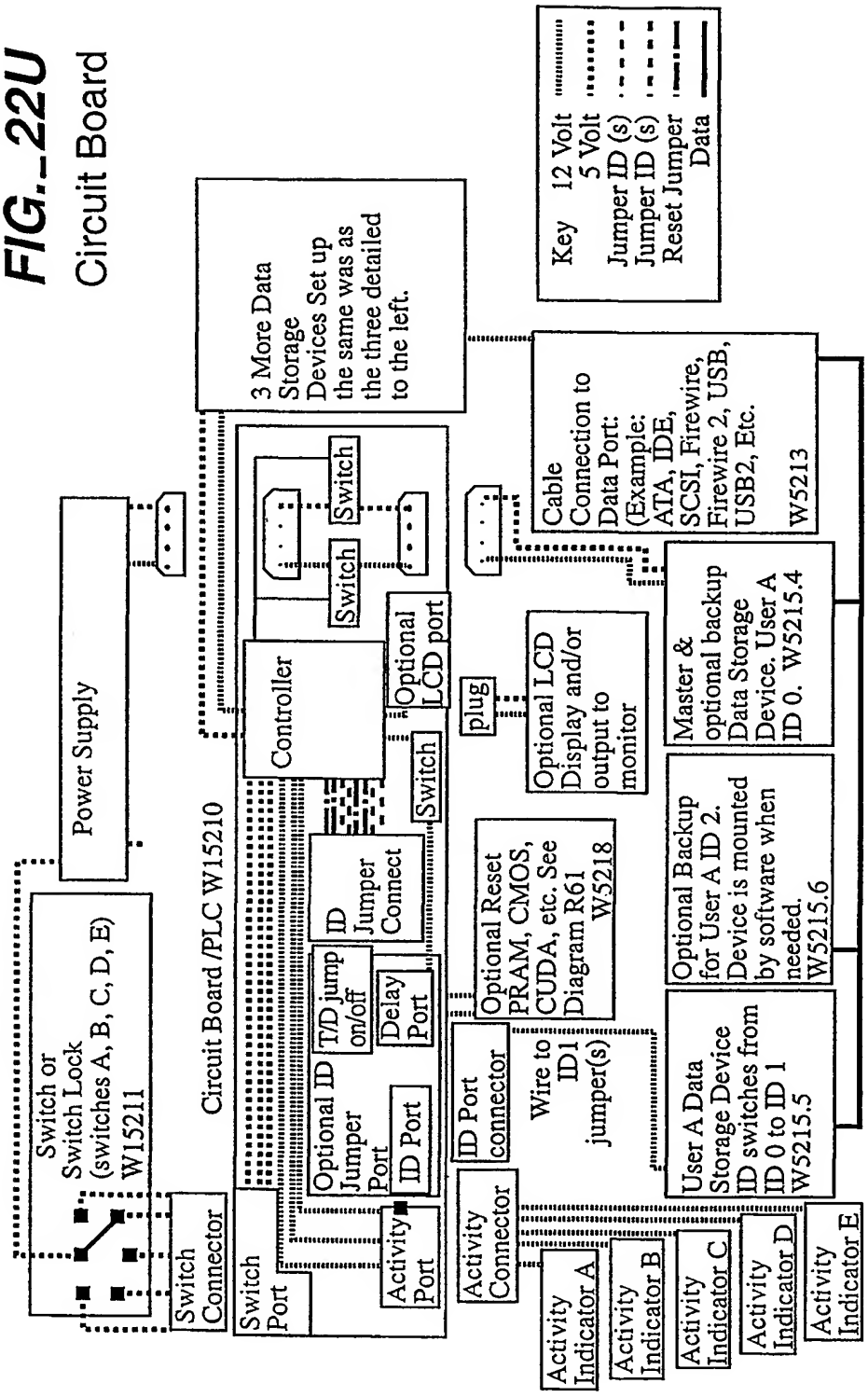
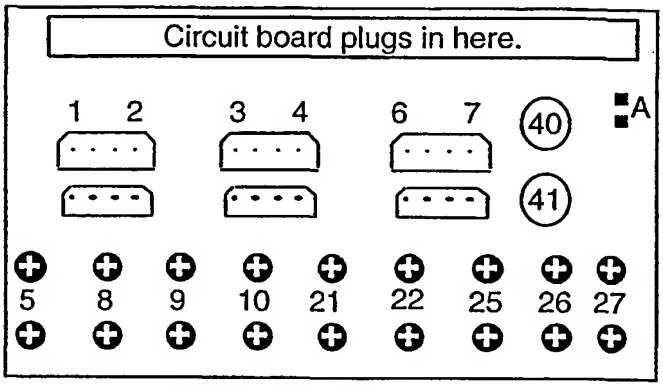

Key 12 Volt
5 Volt
Jumper ID (s)
Jumper ID (s)
Reset Jumper
Data
Circuit Board
Power Supply
Switch or Switch Lock (switches A, B, C) W15211
Switch Connector
Circuit Board /PLC W15210
Switch Port
Activity Port
Optional ID Jumper Port
ID Port
T/D jump on/off
ID Jumper Connect
Delay Port
Controller
Switch
Switch
Optional LCD port
Switch
plug
LCD Display and/or output to monitor
Cable Connection to Data Port: (Example: ATA, IDE, SCSI, Firewire, Firewire 2, USB, USB2, Etc.
W5213
ID O
Master Data Storage Device User A ID 0 W5215.4
ID selection Cable
Backup for User A ID 2. Device is mounted by software when needed. W5215.6
User A Data Storage Device ID switches from ID 0 to ID 1 W5215.5
ID 0/ID 1 switch
Optional Reset PRAM, CMOS, CUDA, etc. See Diagram R61 W5218
ID Port connector
Activity Connector
Activity Indicator A
Activity Indicator B
Activity Indicator C

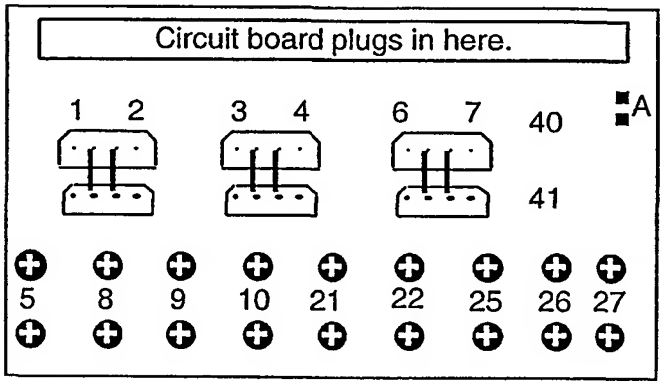
FIG._22U
Circuit Board



Circuit Board Single-User & Repair



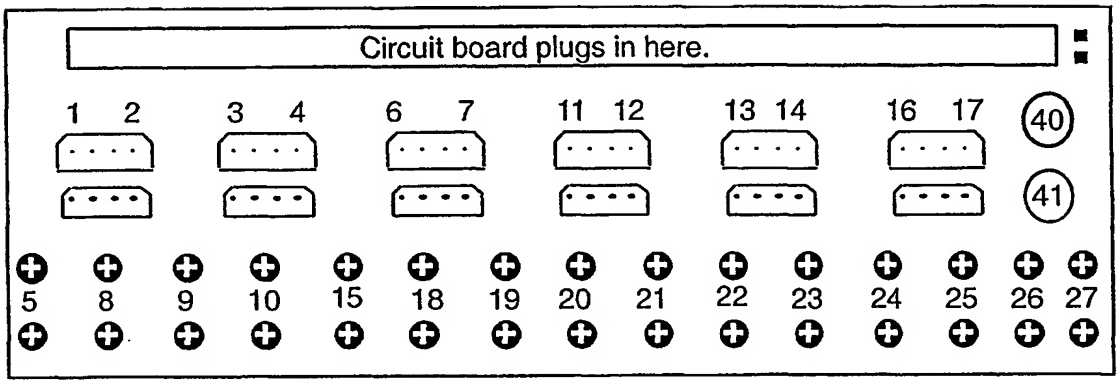
Socket Top View



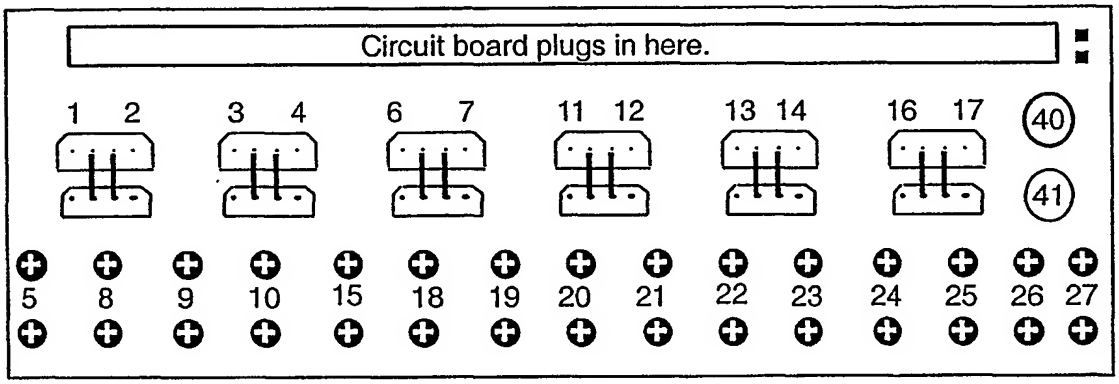
Bottom View of Power Sockets--Jumpers on bottom of socket neutral terminals

FIG._22V

Multi-User & Repair

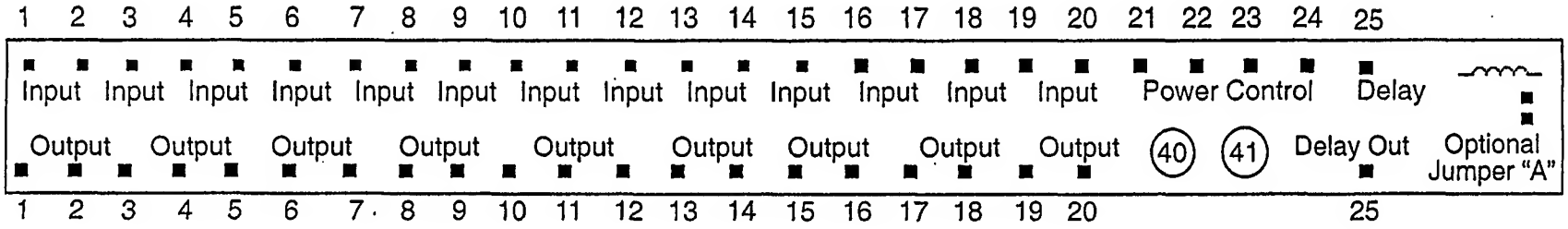


Socket Top View



Bottom View of Power Sockets--Jumpers on bottom of socket neutral terminals

FIG._22W



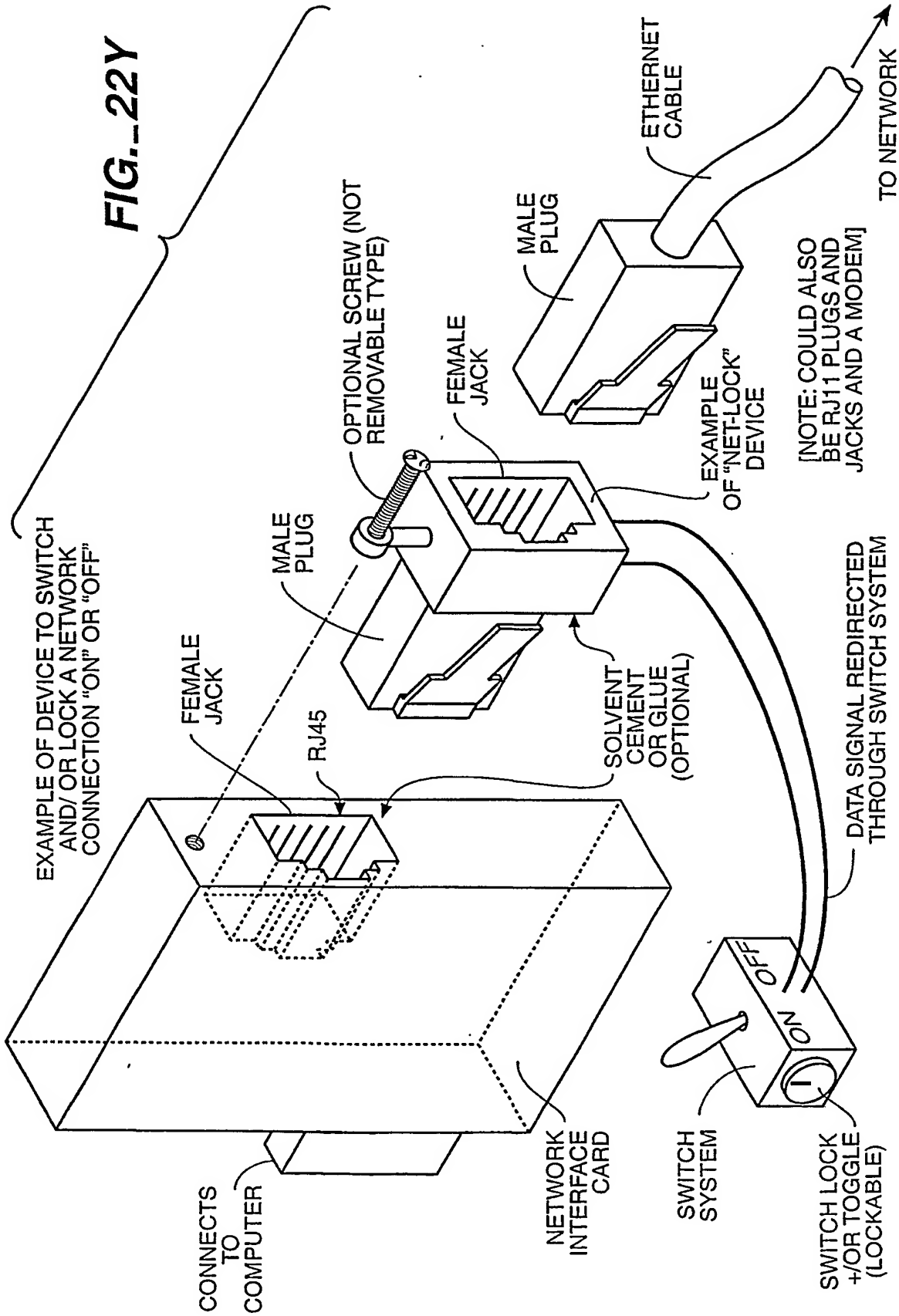
1) circuit 1
2) circuit 2
3) circuit 3
4) circuit 4
5) circuit 5
6) circuit 6
7) circuit 7
8) circuit 8
9) circuit 9
10) circuit 10
11) circuit 11
12) circuit 12
13) circuit 13
14) circuit 14
15) circuit 15
16) circuit 16
17) circuit 17
18) circuit 18
19) circuit 19
20) circuit 20
21) Power Control Indicator #21
22) Power Control Indicator #22
23) Power Control Indicator #23
24) Power Control Indicator #24
25) time delay circuit
40) delay start time for delay circuit
41) stop time for delay circuit

BURPMU may need 4/22/01: Circuit board 7/15/2000

Socket Top View

FIG._22X

FIG._22Y


EXAMPLE OF DEVICE TO SWITCH AND/ OR LOCK A NETWORK CONNECTION "ON" OR "OFF"
FEMALE JACK
RJ45
MALE PLUG
OPTIONAL SCREW (NOT REMOVABLE TYPE)
FEMALE JACK
SOLVENT CEMENT OR GLUE (OPTIONAL)
MALE PLUG
ETHERNET CABLE
TO NETWORK
EXAMPLE OF "NET-LOCK" DEVICE
[NOTE: COULD ALSO BE RJ11 PLUGS AND JACKS AND A MODEM]
DATA SIGNAL REDIRECTED THROUGH SWITCH SYSTEM
OFF
ON
CONNECTS TO COMPUTER
NETWORK INTERFACE CARD
SWITCH SYSTEM
SWITCH LOCK +/OR TOGGLE (LOCKABLE)

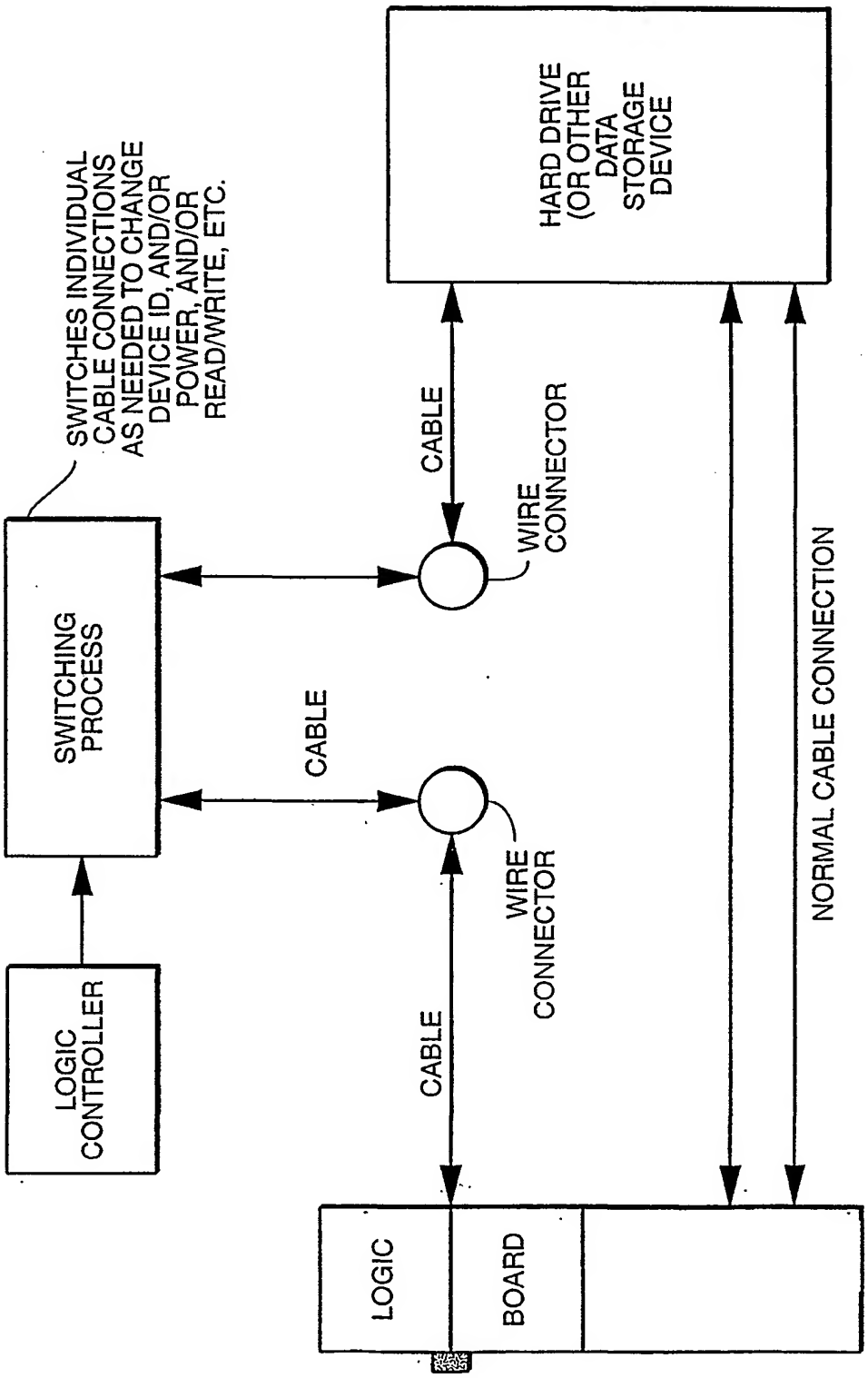

HARD DRIVE (OR OTHER DATA STORAGE DEVICE
SWITCHES INDIVIDUAL CABLE CONNECTIONS AS NEEDED TO CHANGE DEVICE ID, AND/OR POWER, AND/OR READ/WRITE, ETC.
CABLE
WIRE CONNECTOR
SWITCHING PROCESS
CABLE
WIRE CONNECTOR
LOGIC CONTROLLER
CABLE
NORMAL CABLE CONNECTION
LOGIC
BOARD

FIG._22Z

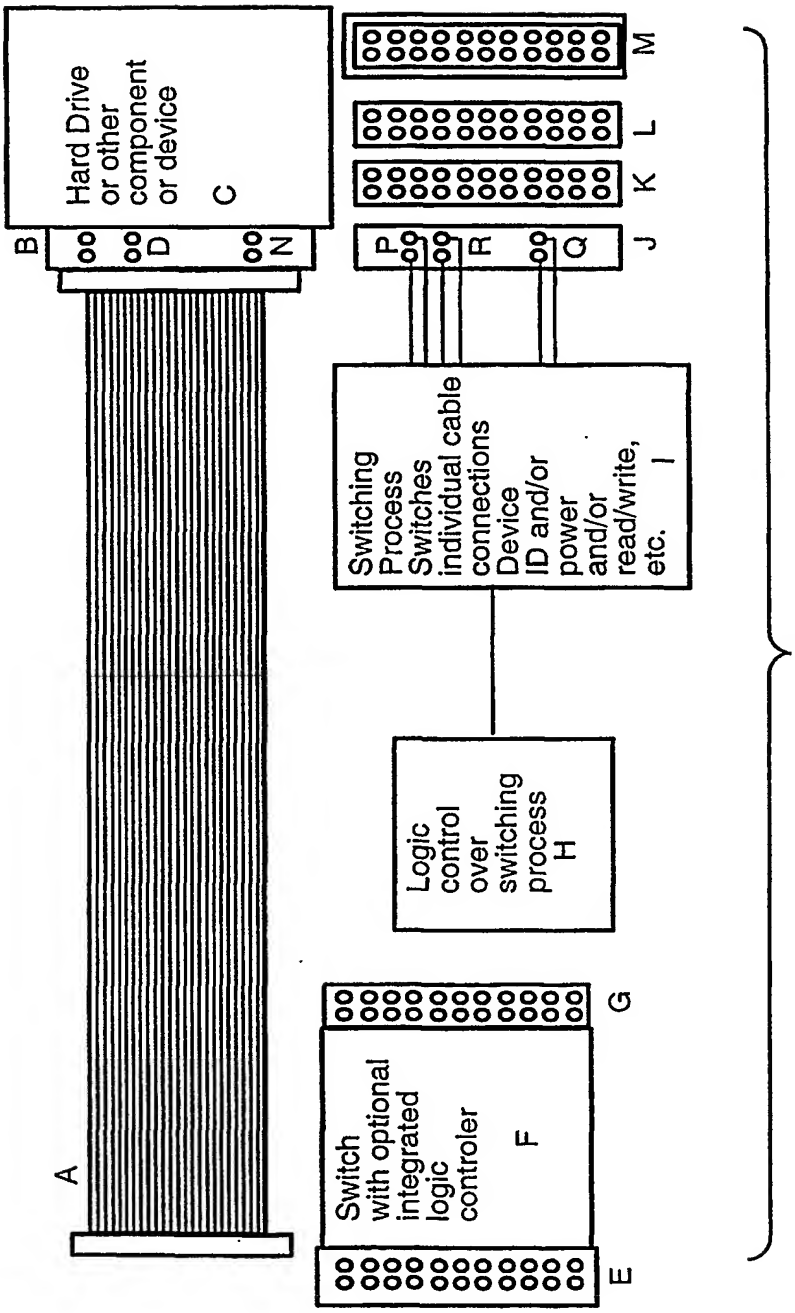

Hard Drive or other component or device C
B
D
N
A
M
L
K
J
P
R
Q
Switching Process Switches individual cable connections Device ID and/or power and/or read/write, etc. I
Logic control over switching process H
Switch with optional integrated logic controler F
G
E

FIG._22Z-1

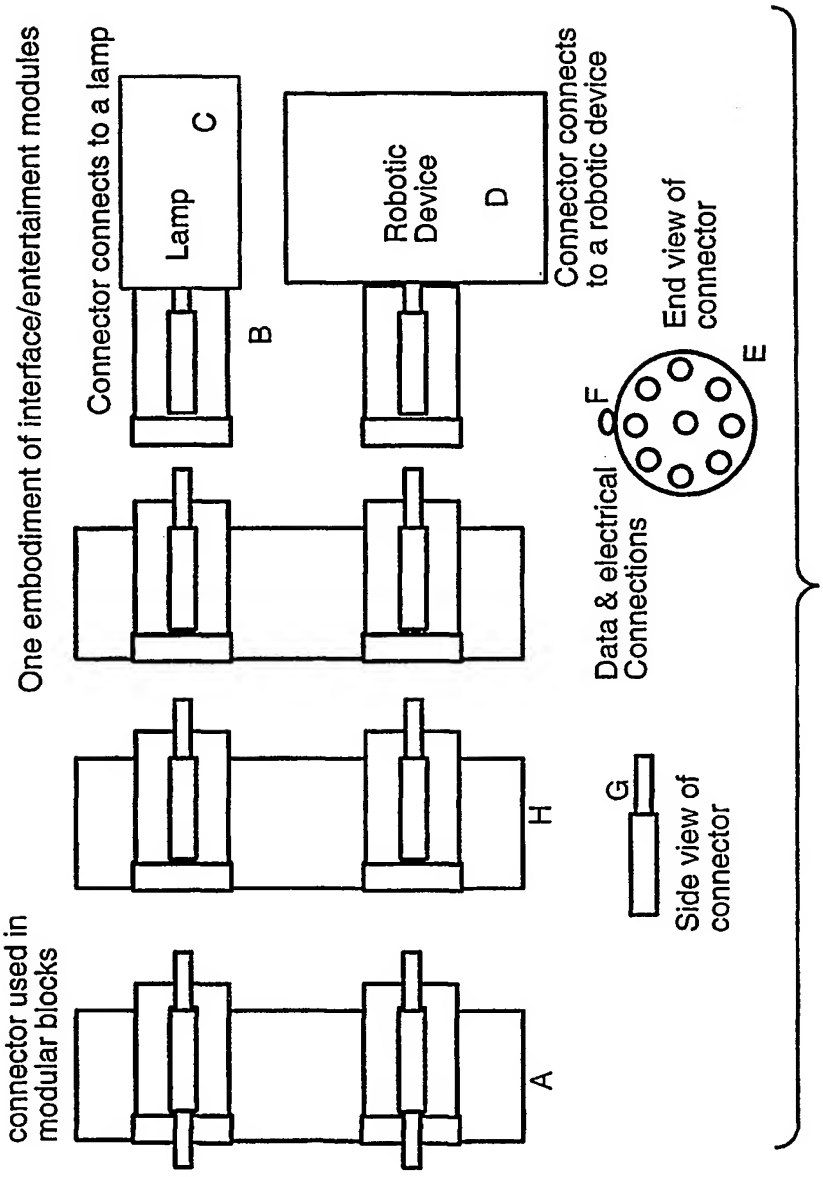

One embodiment of interface/entertaiment modules
connector used in modular blocks
A
H
Connector connects to a lamp
Lamp
C
B
Robotic Device
D
Connector connects to a robotic device
Data & electrical Connections
F
End view of connector
E
G
Side view of connector

FIG._22Z-2

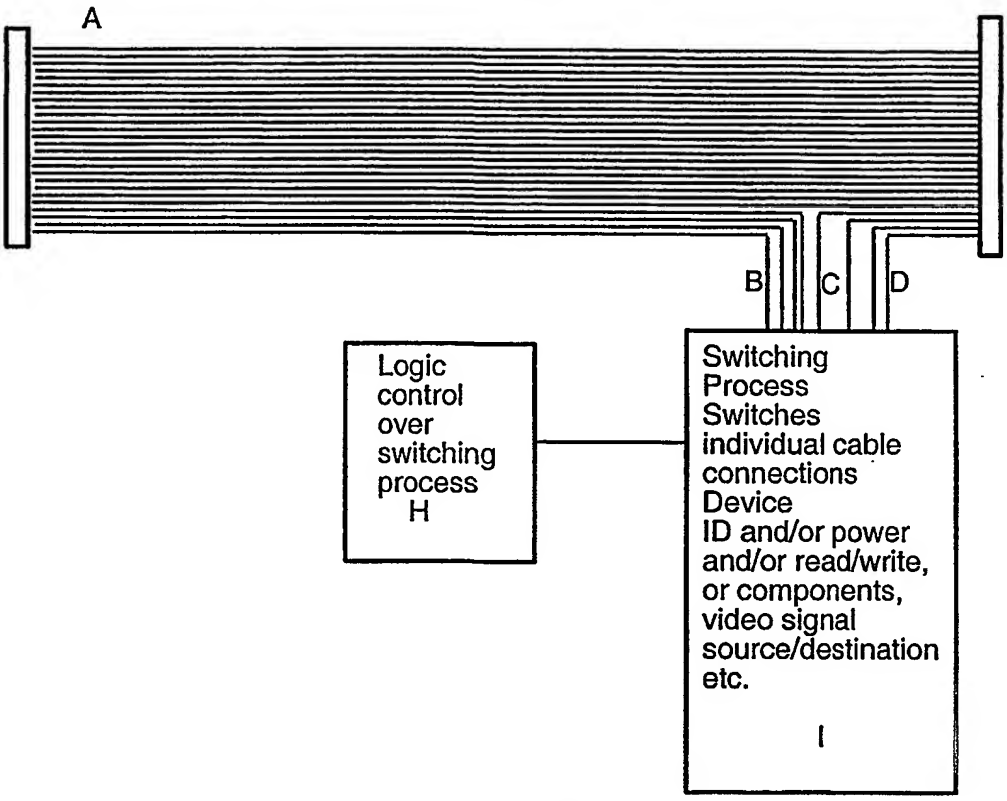

A
B
C
D
Logic control over switching process H
Switching Process Switches individual cable connections Device ID and/or power and/or read/write, or components, video signal source/destination etc.
I

FIG._22Z-3

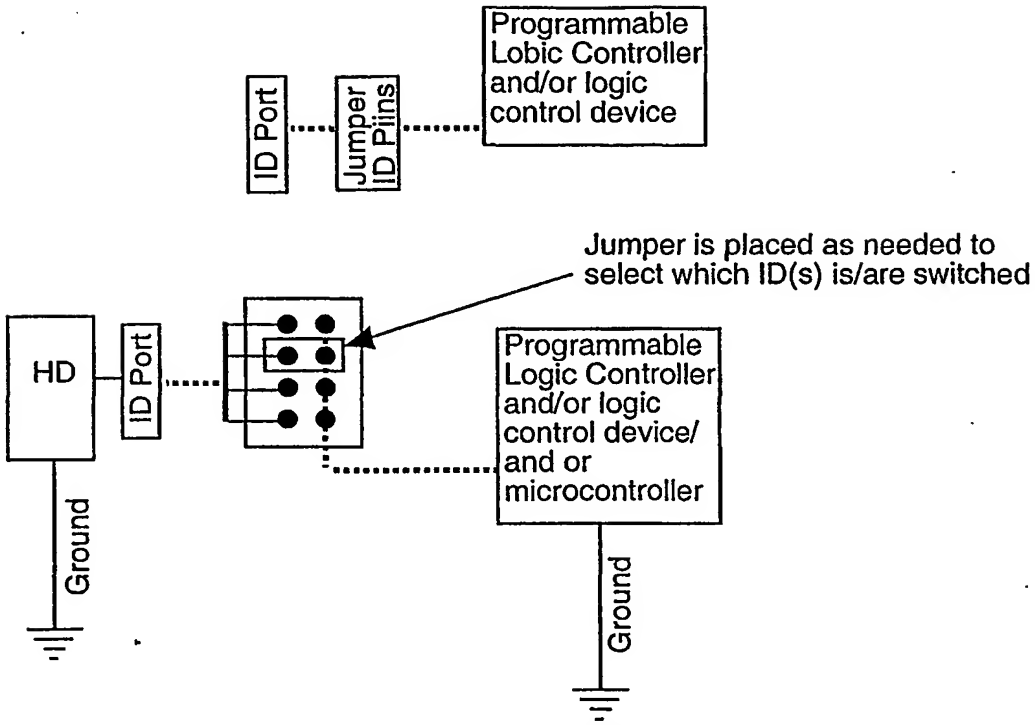

Programmable
Lobic Controller
and/or logic
control device
ID Port
Jumper
ID Pins
Jumper is placed as needed to
select which ID(s) is/are switched
HD
ID Port
Programmable
Logic Controller
and/or logic
control device/
and or
microcontroller
Ground
Ground
Pink are common on the circuit board. they connect to 1 IO pin
of the microcontoller Black connect to A0 A1 A2 A3 on hard drive.

FIG._23A

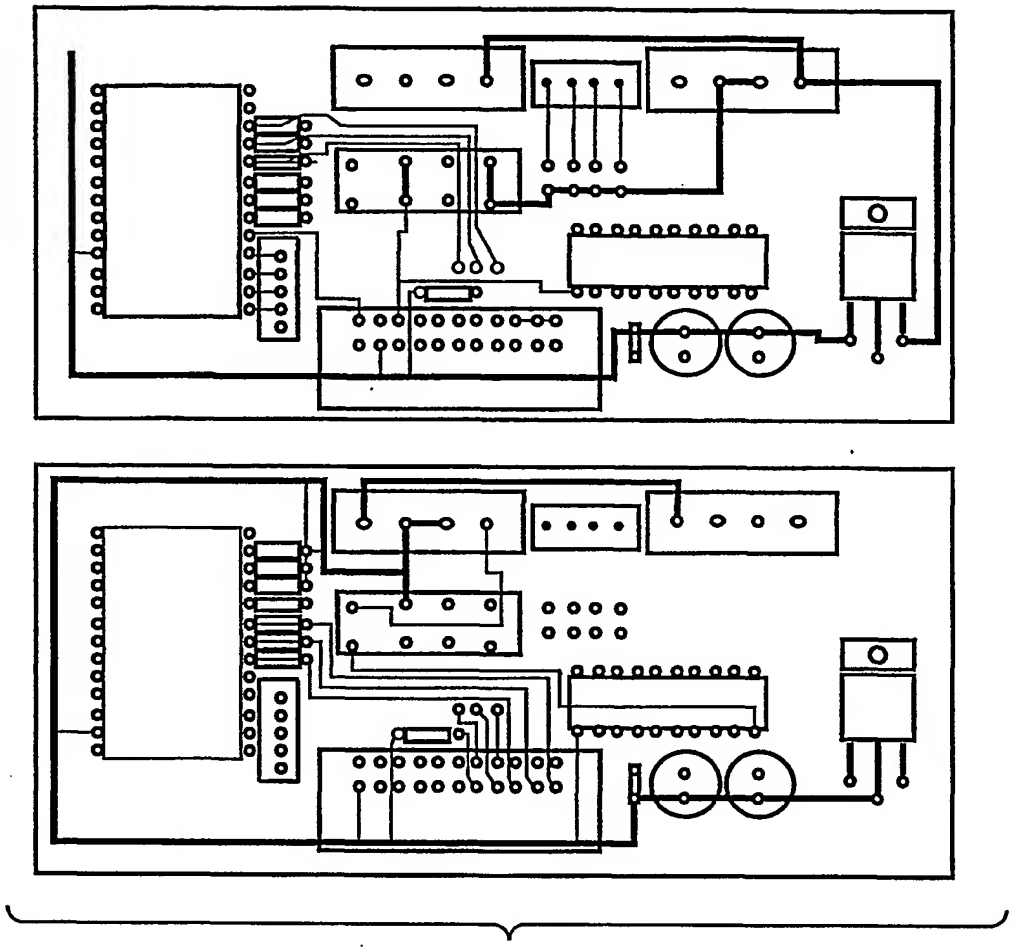


FIG._23B

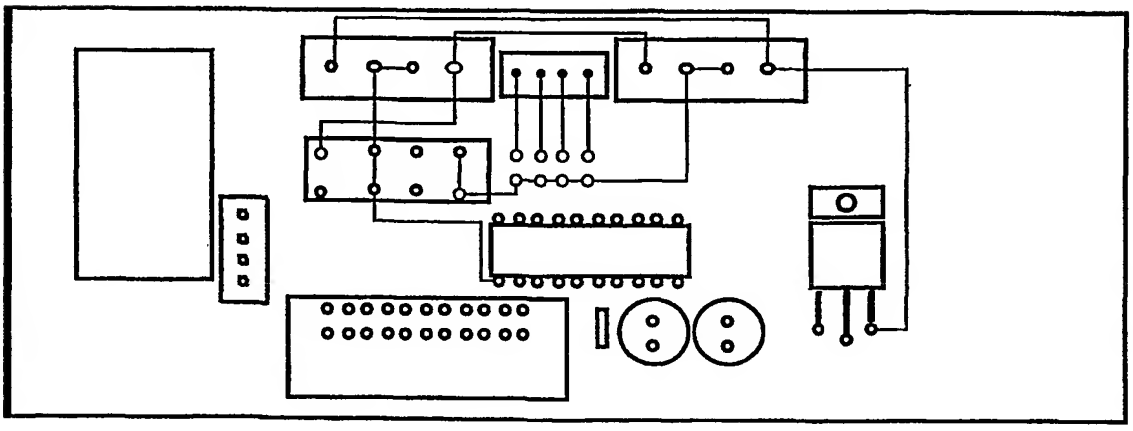


FIG._23C

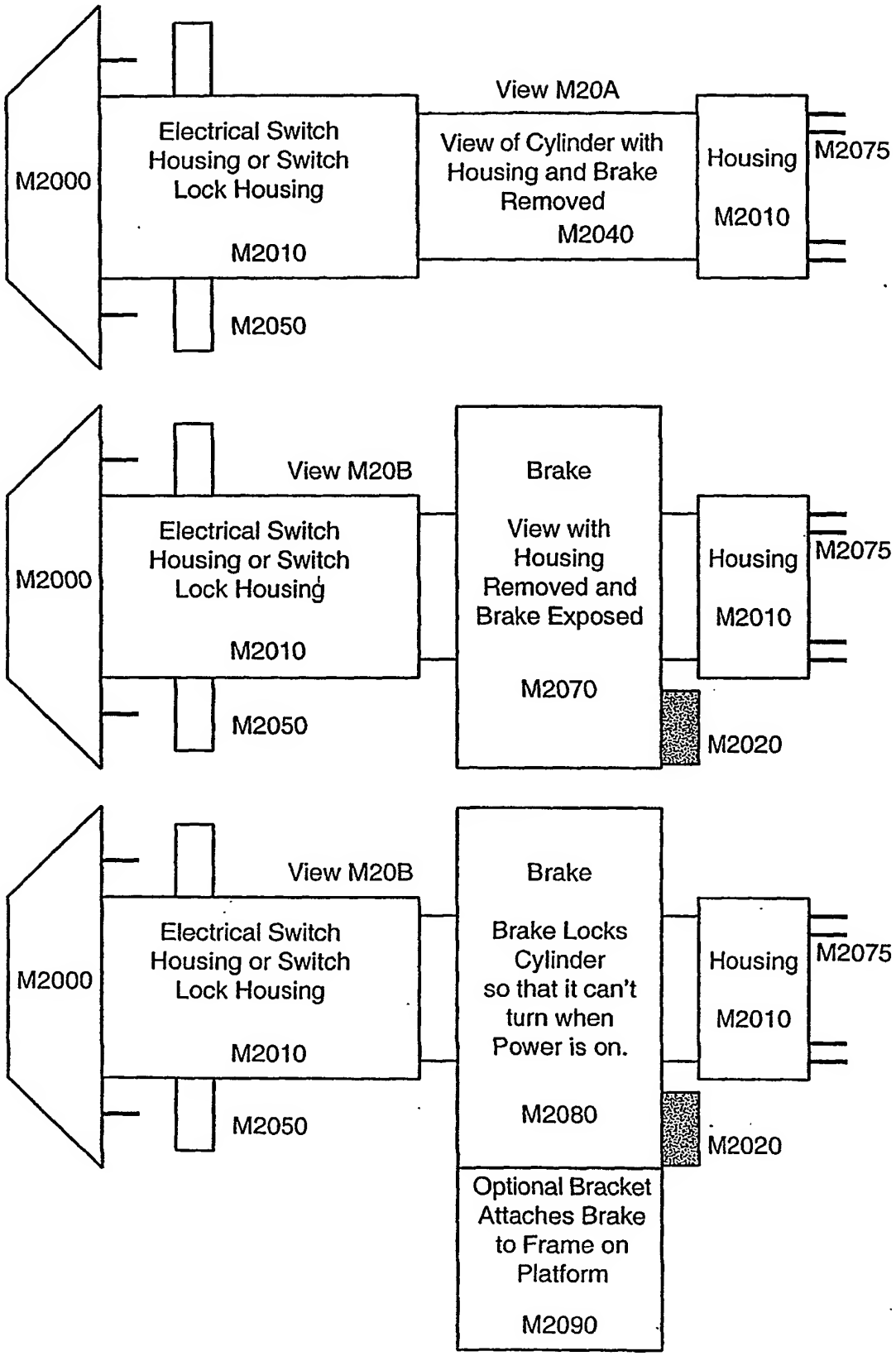

M2000
Electrical Switch Housing or Switch Lock Housing
M2010
M2050
View M20A
View of Cylinder with Housing and Brake Removed
M2040
Housing
M2010
M2075
M2000
View M20B
Electrical Switch Housing or Switch Lock Housing
M2010
M2050
Brake
View with Housing Removed and Brake Exposed
M2070
M2020
Housing
M2010
M2075
M2000
View M20B
Electrical Switch Housing or Switch Lock Housing
M2010
M2050
Brake
Brake Locks Cylinder so that it can't turn when Power is on.
M2080
M2020
Optional Bracket Attaches Brake to Frame on Platform
M2090
Housing
M2010
M2075

FIG._24A

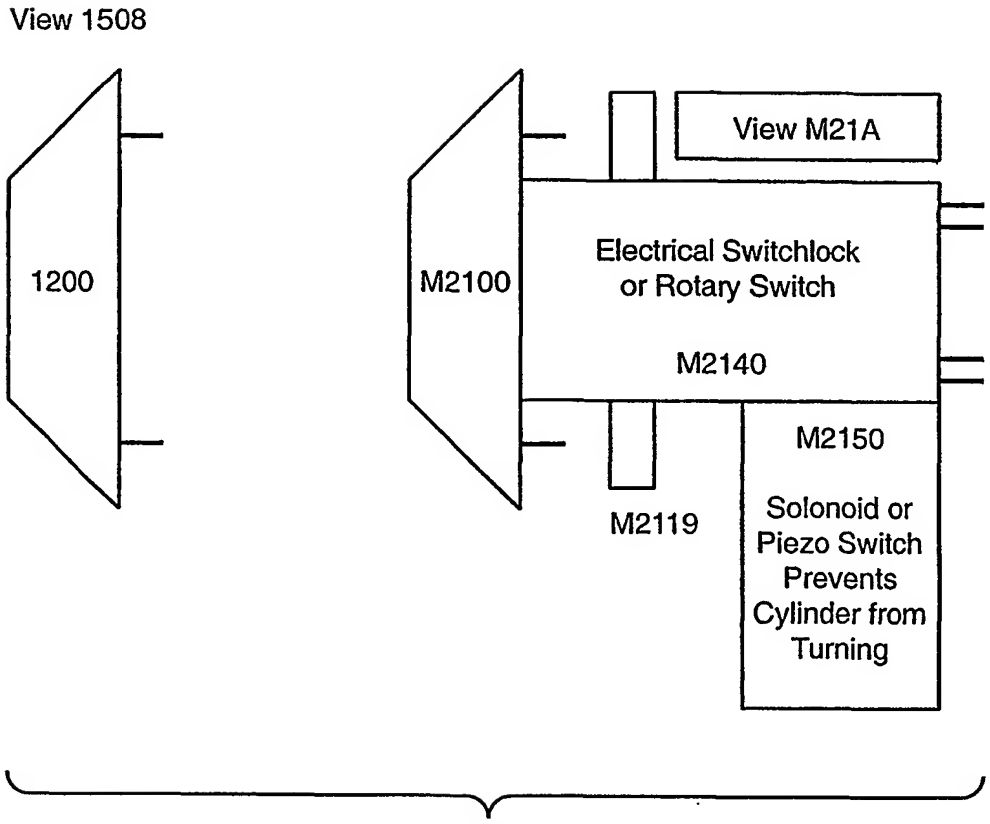

View 1508
1200
M2100
View M21A
Electrical Switchlock
or Rotary Switch
M2140
M2150
M2119
Solonoid or
Piezo Switch
Prevents
Cylinder from
Turning

FIG._24B

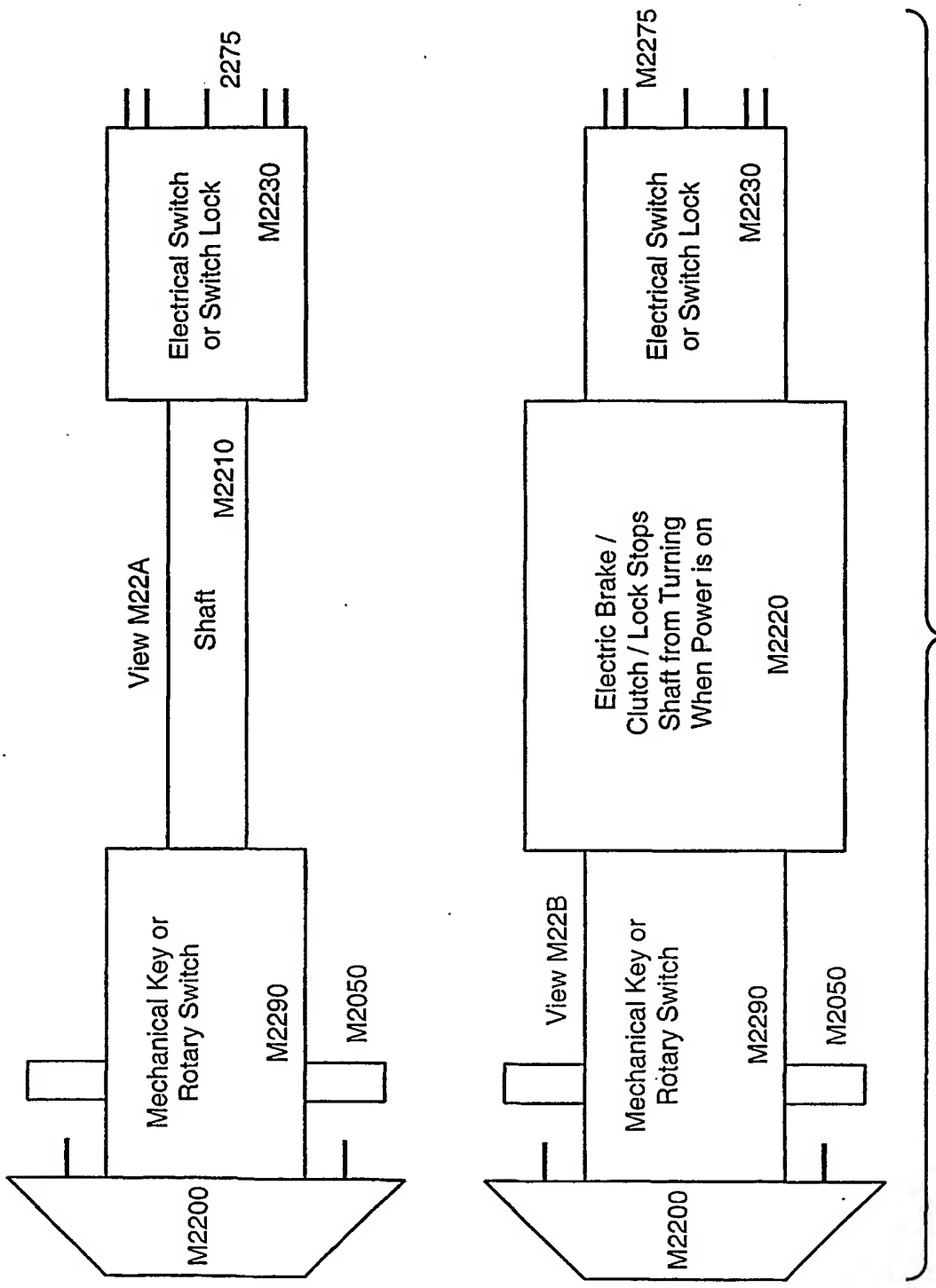

M2200
Mechanical Key or Rotary Switch
M2290
M2050
View M22A
Shaft
M2210
Electrical Switch or Switch Lock
M2230
2275
View M22B
Electric Brake / Clutch / Lock Stops Shaft from Turning When Power is on
M2220
M2275

FIG._24C

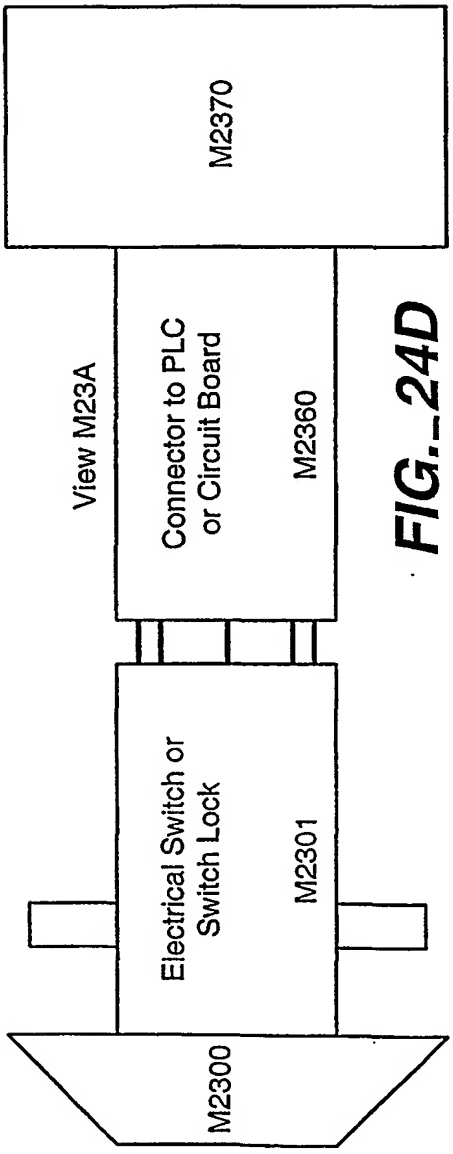


FIG._24D

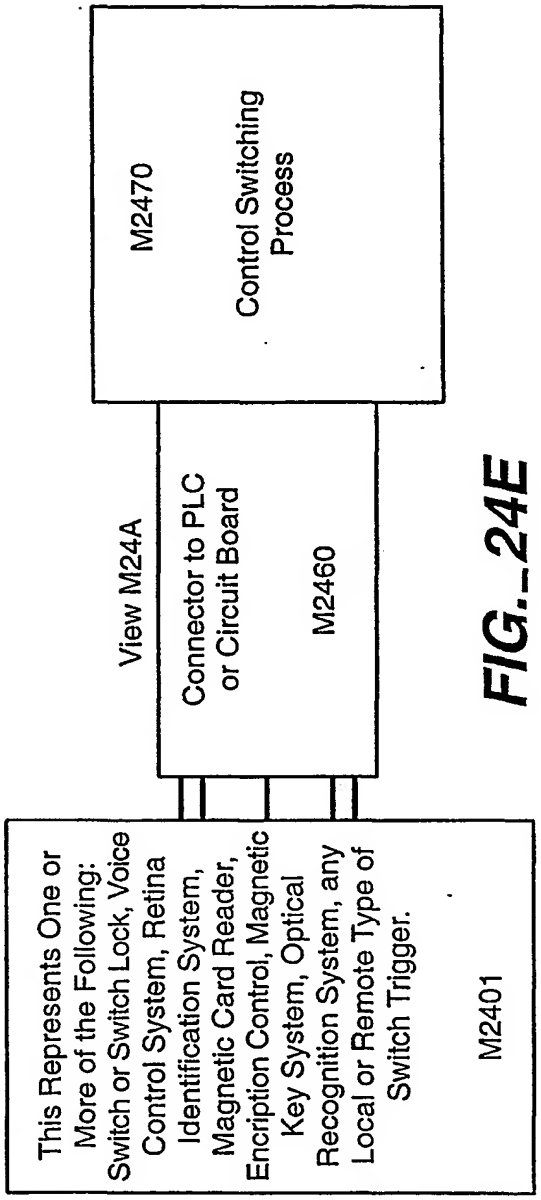


FIG._24E

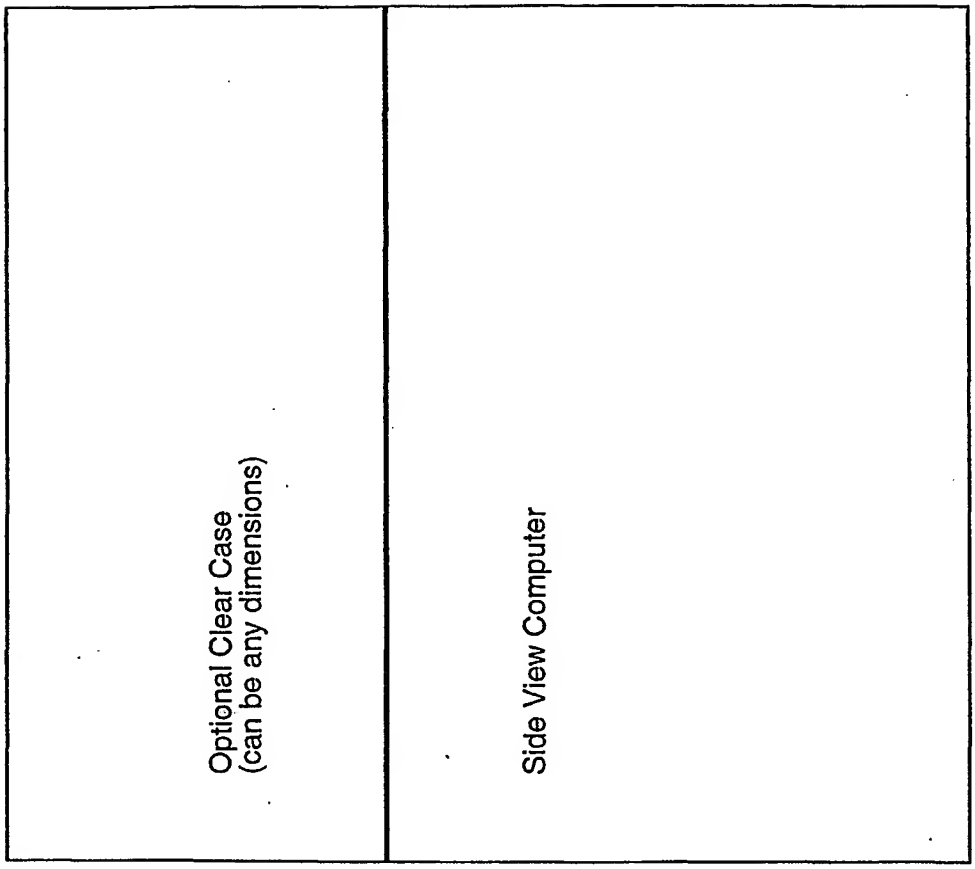


FIG._25B

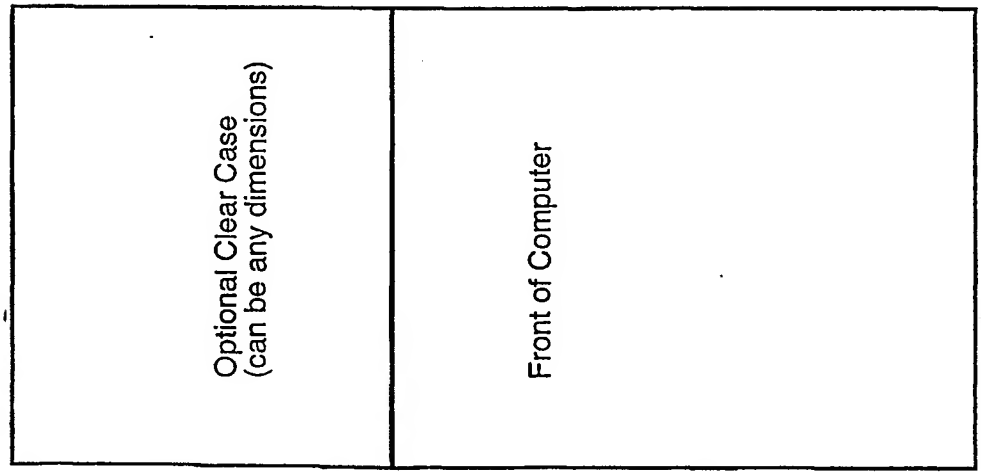


FIG._25A

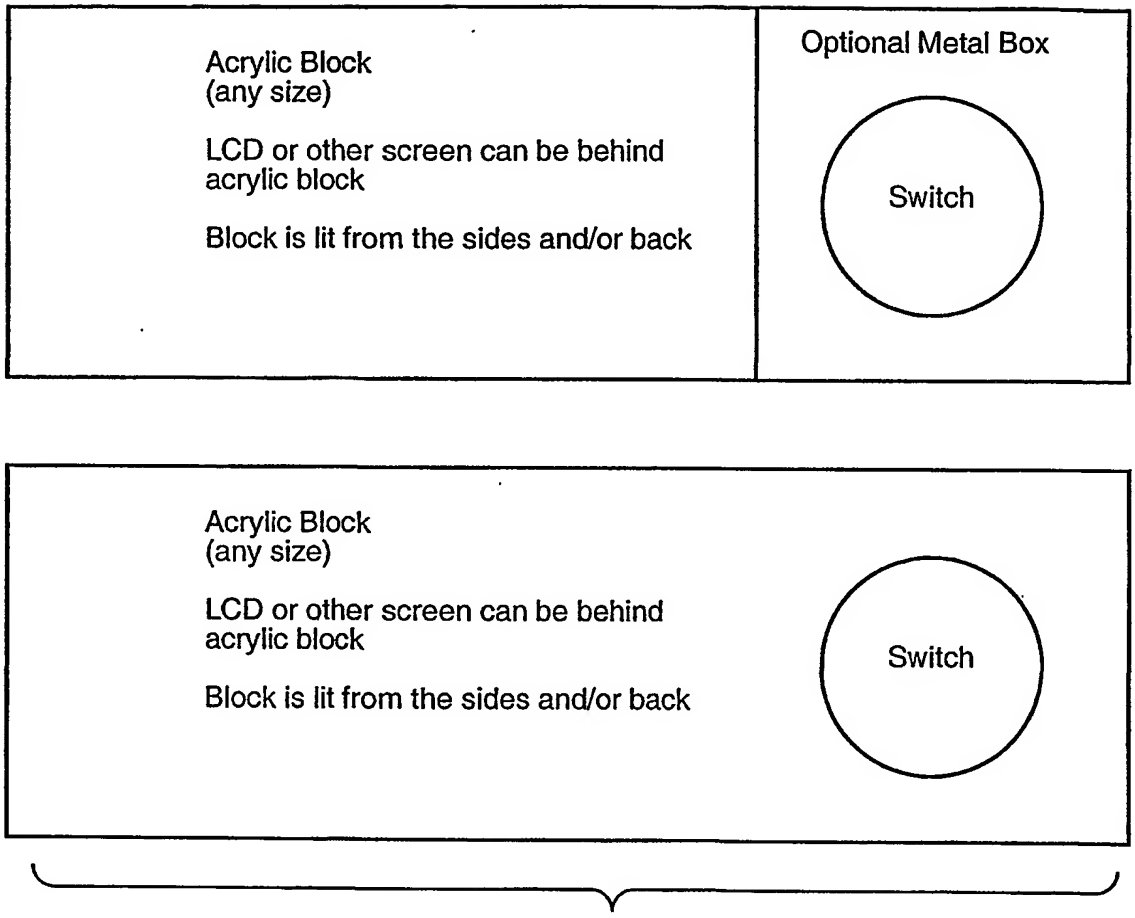
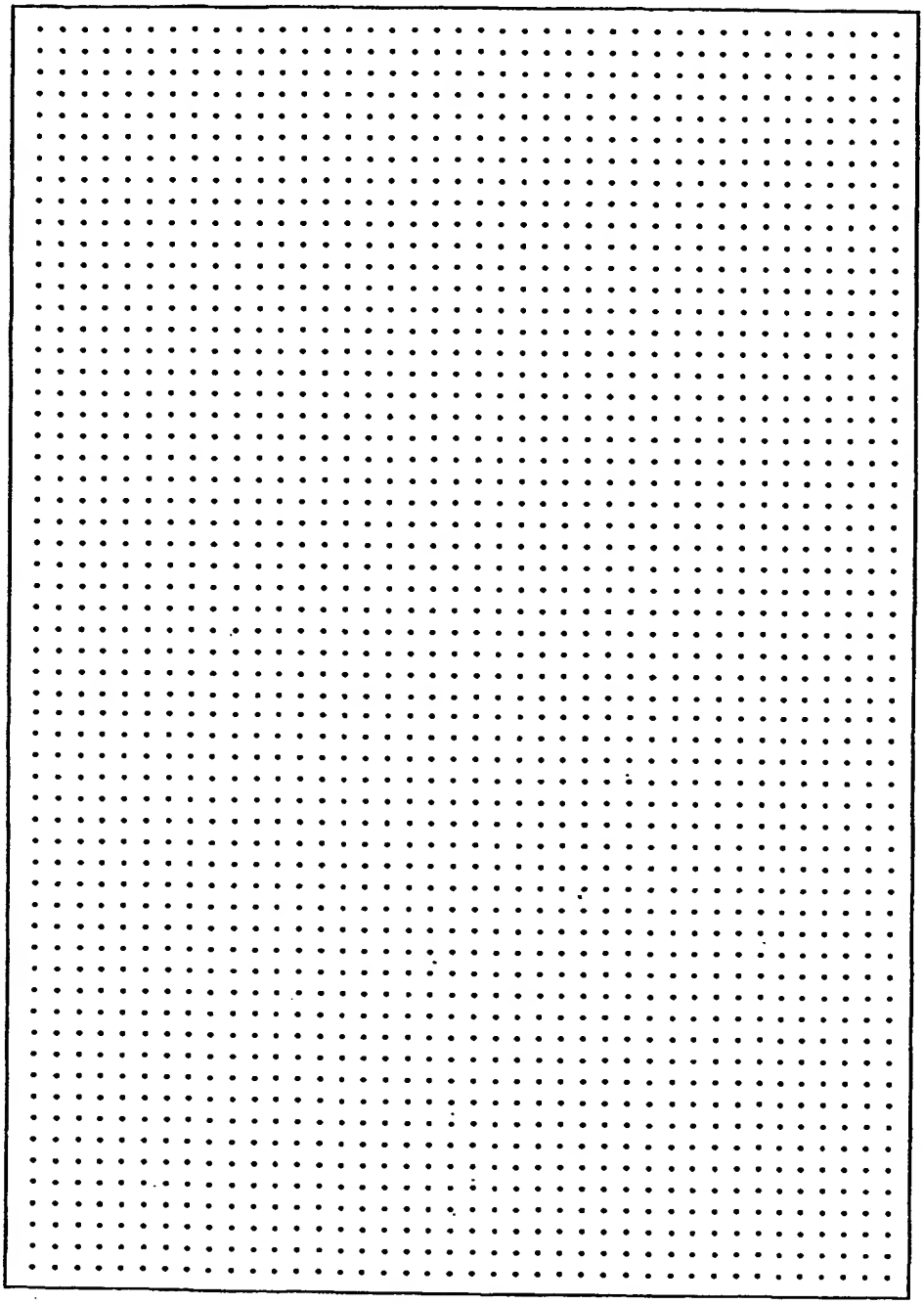

Acrylic Block
(any size)
LCD or other screen can be behind acrylic block
Block is lit from the sides and/or back
Optional Metal Box
Switch
Acrylic Block
(any size)
LCD or other screen can be behind acrylic block
Block is lit from the sides and/or back
Switch

FIG._25C

Optional Top and/or Sides and/or Back of Computer/Computing Device/Peripheral Device



Optional Lip That Optional Case fits over

FIG._25D

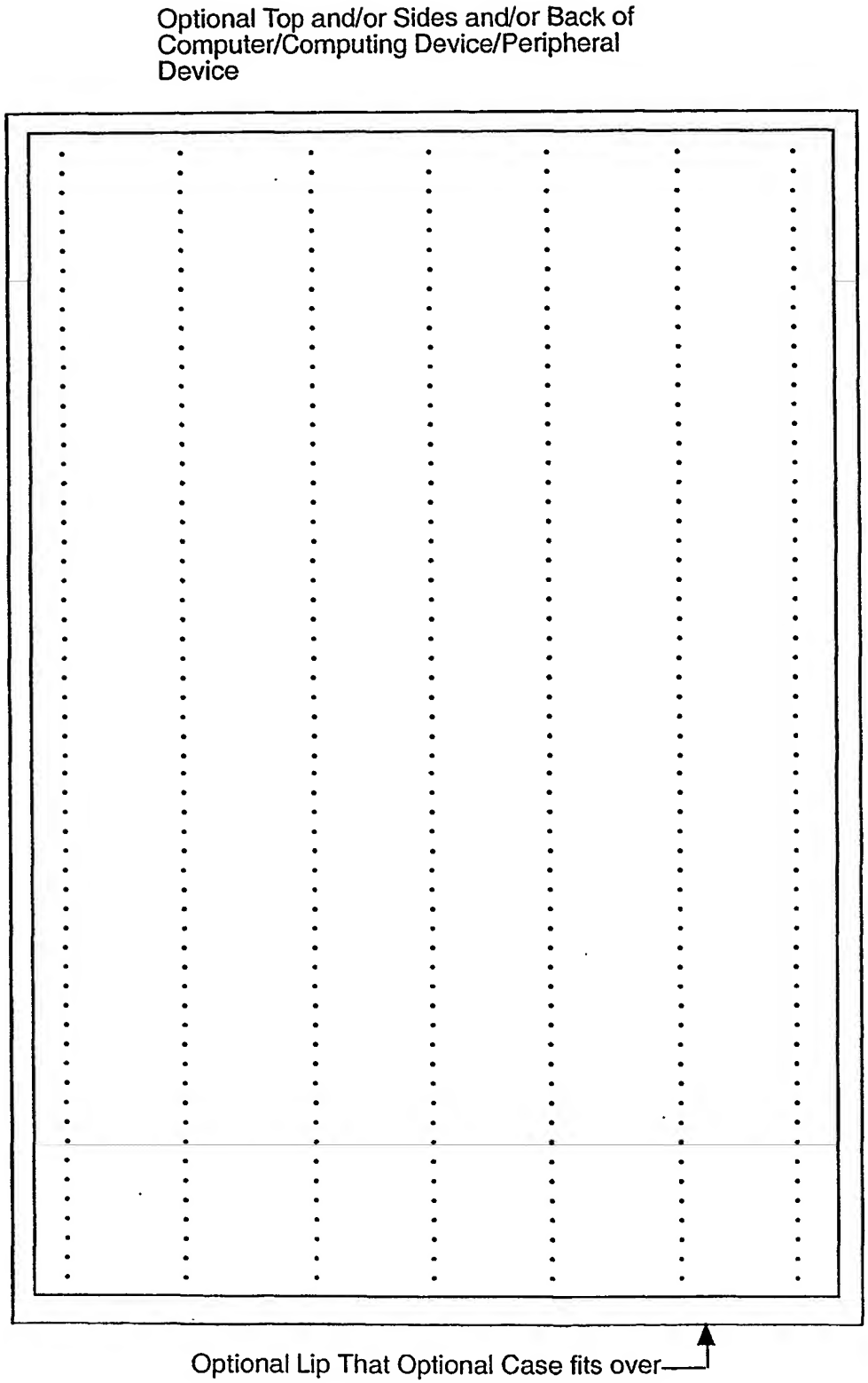

Optional Top and/or Sides and/or Back of Computer/Computing Device/Peripheral Device
Optional Lip That Optional Case fits over

FIG._25E

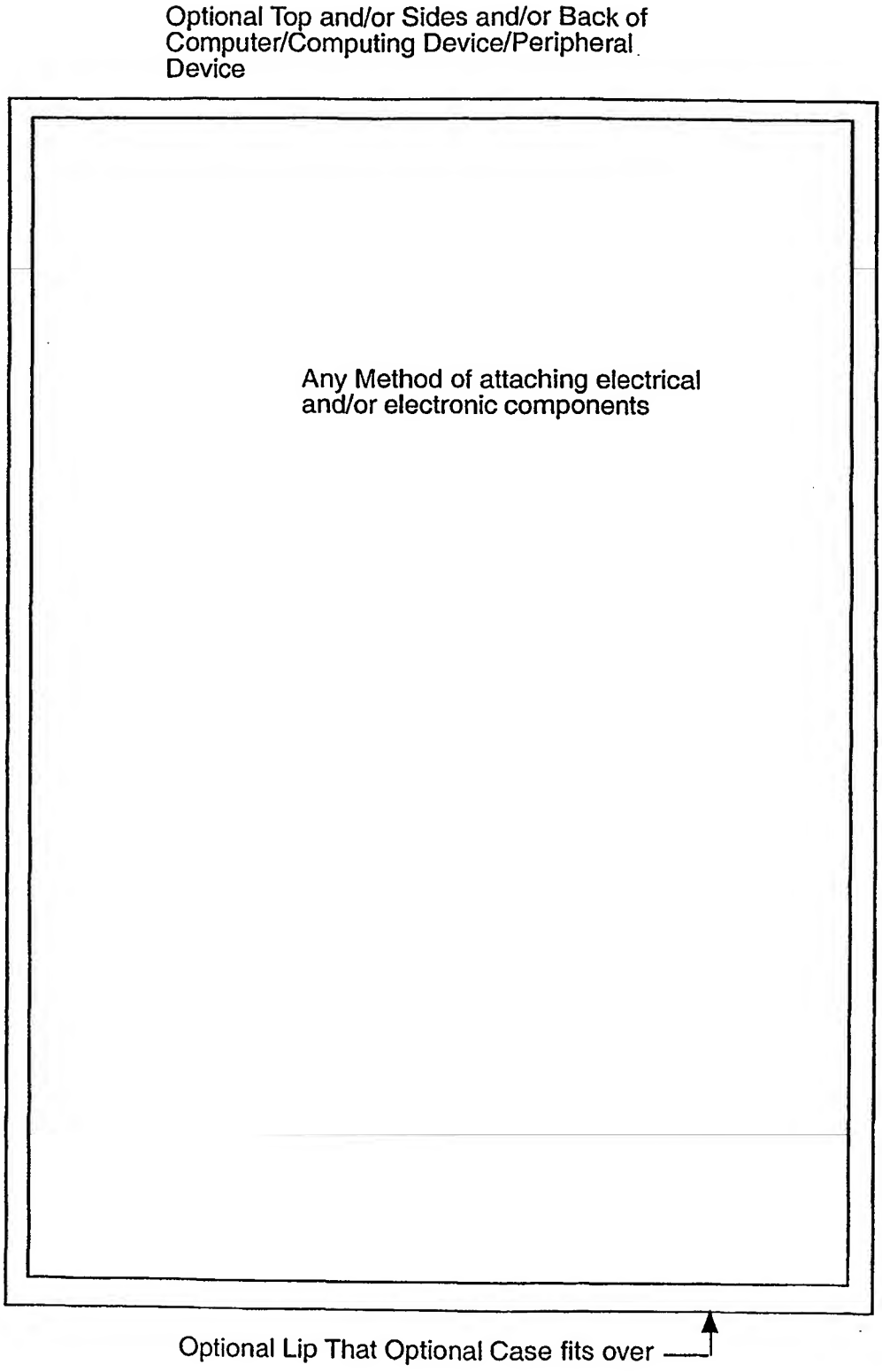

Optional Top and/or Sides and/or Back of Computer/Computing Device/Peripheral Device
Any Method of attaching electrical and/or electronic components
Optional Lip That Optional Case fits over

FIG._25F

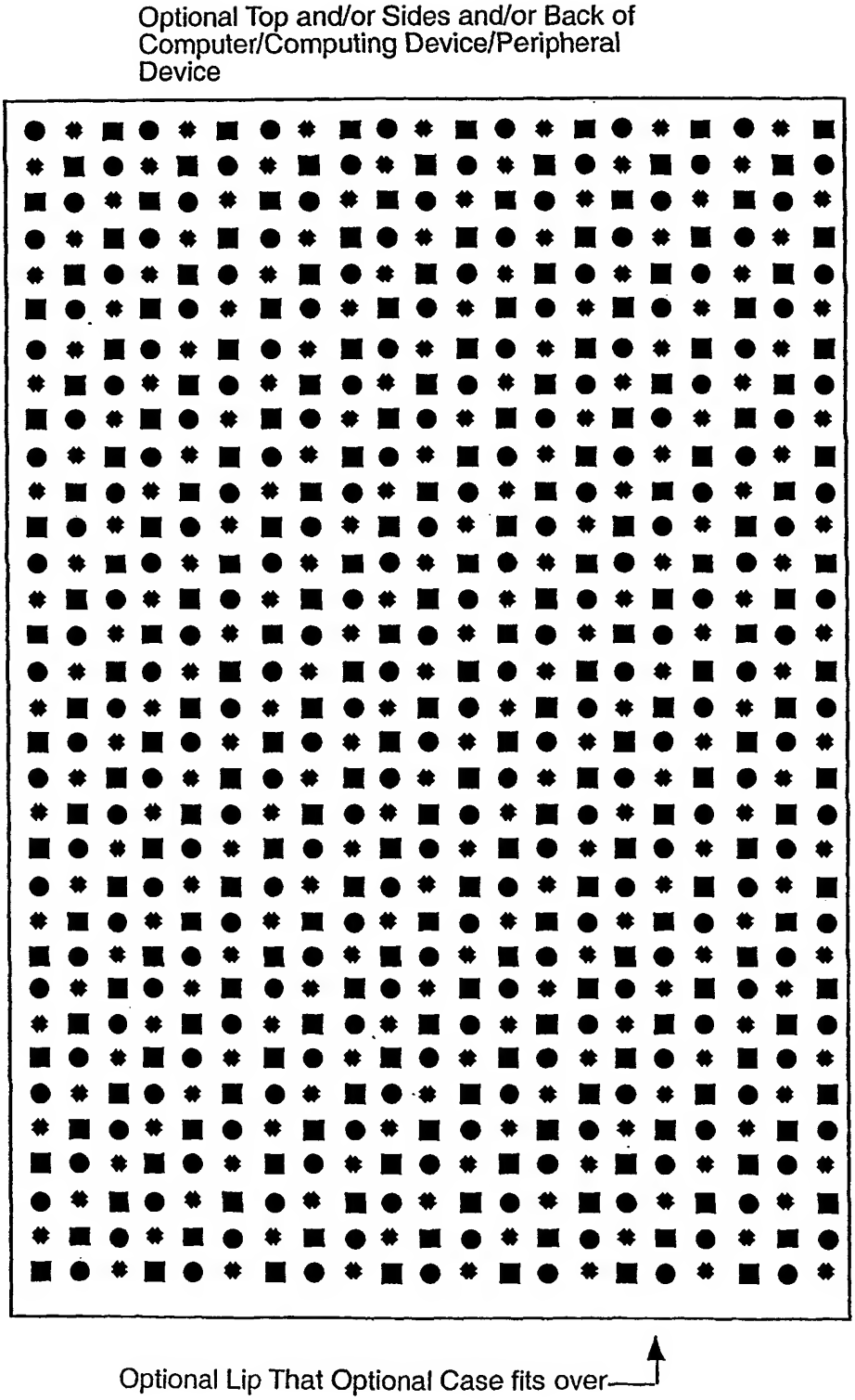

Optional Top and/or Sides and/or Back of Computer/Computing Device/Peripheral Device
Optional Lip That Optional Case fits over

FIG._25G

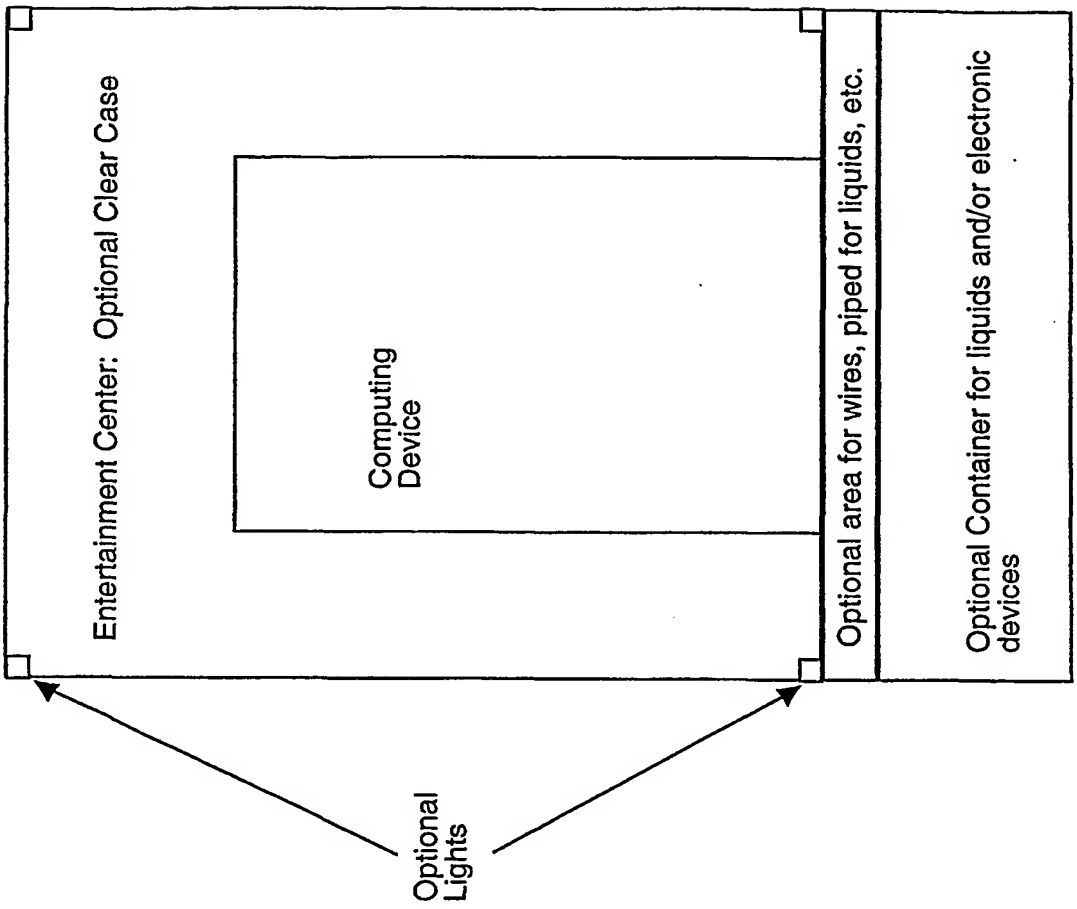


FIG._25I

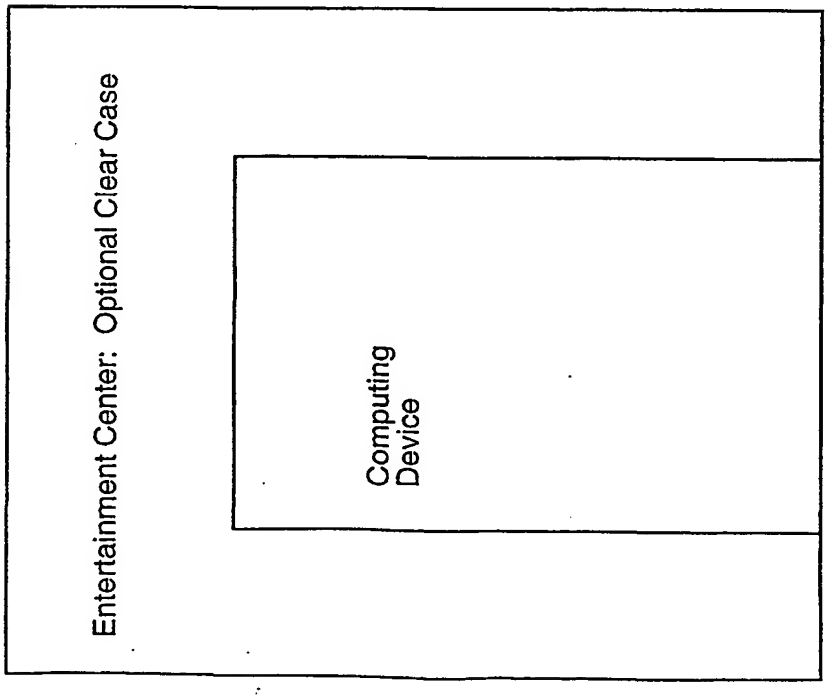
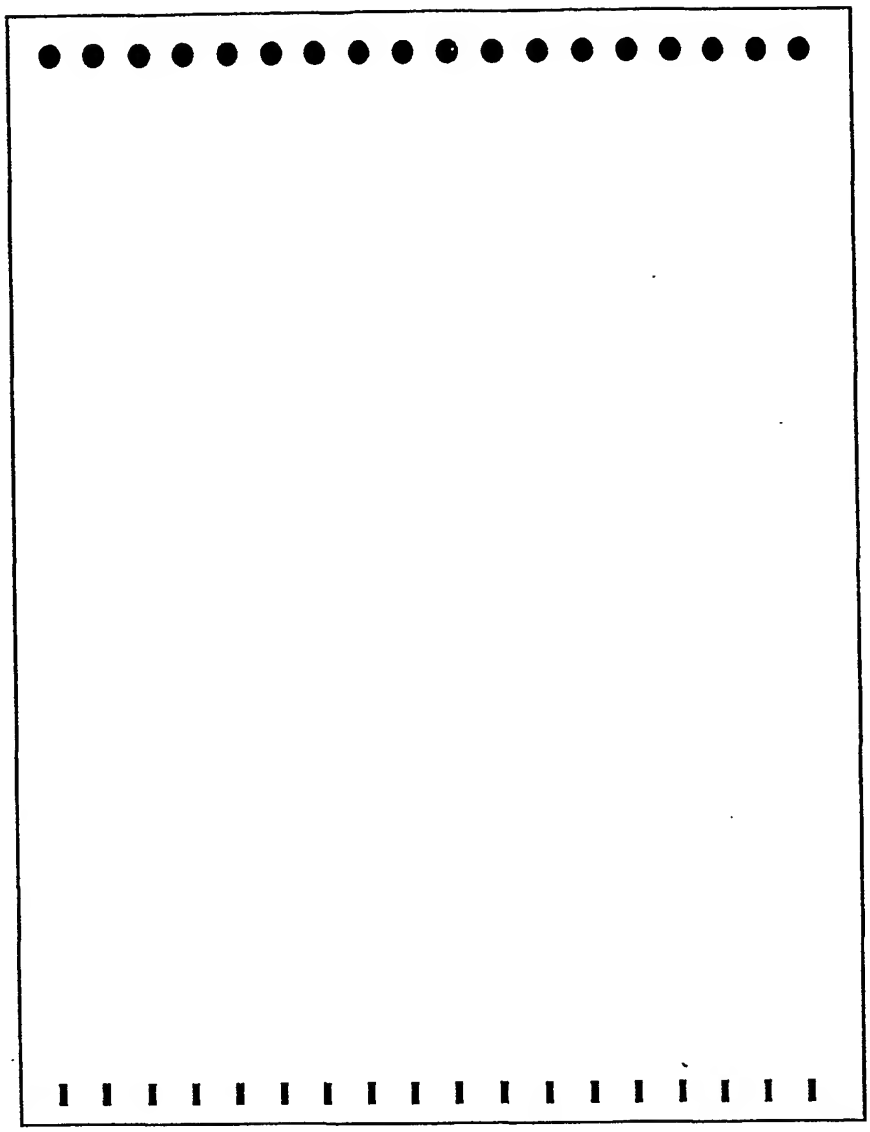
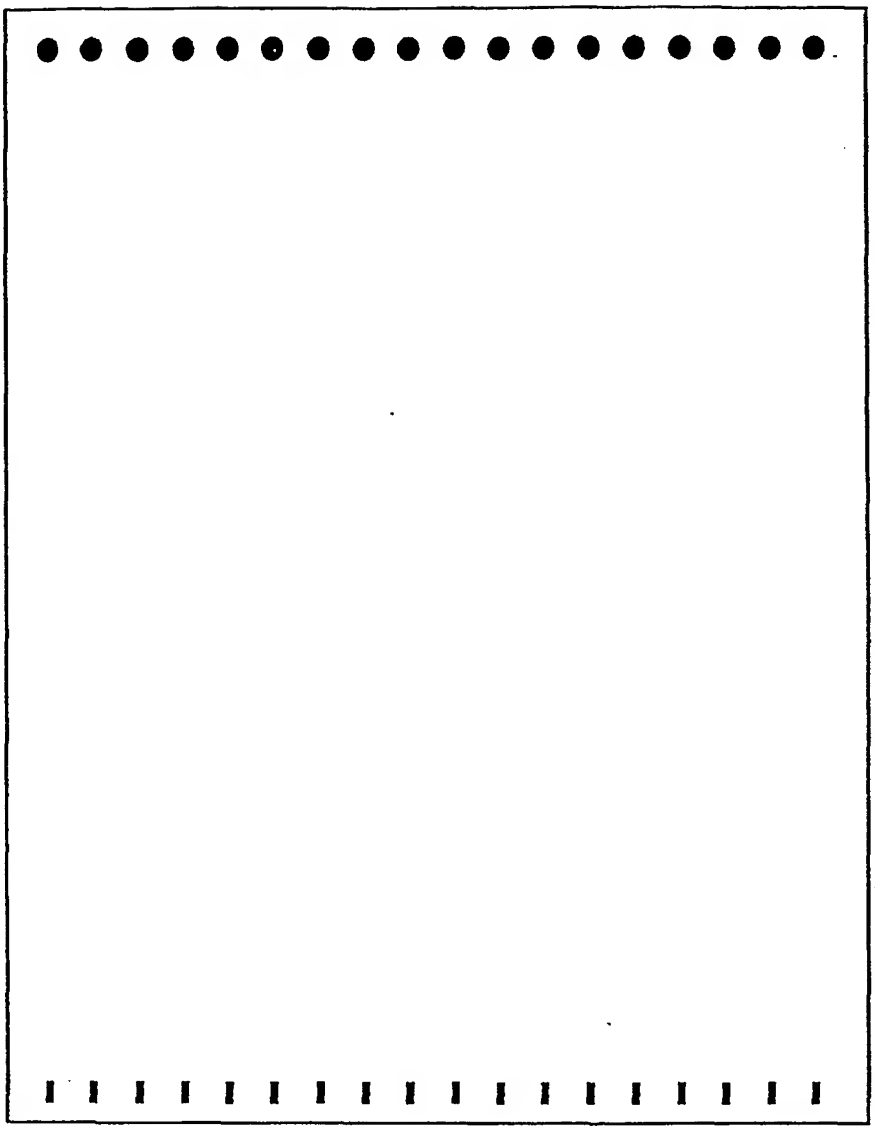


FIG._25H



Optional Top and/or sides. Circuit board and/or device connector board plugs into power sources on top of computing device.

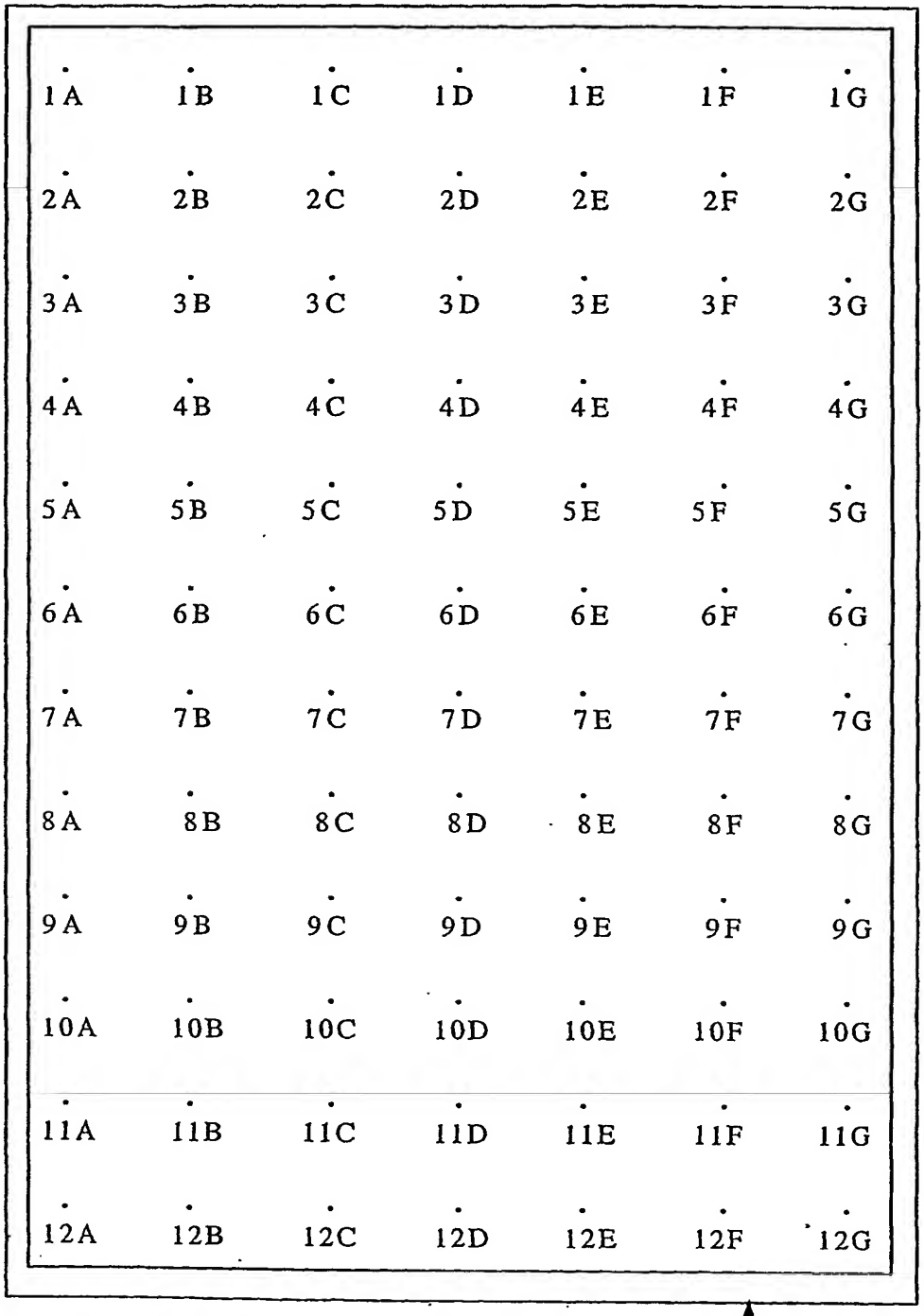
FIG._25J



Optional Example circuit board (bottom View) and/or device connector board plugs. It plugs into power sources on top and/or sides of computing device. This is a functional design that only shows connections to power sources from computing device.

FIG._25K

Optional Top and/or Sides and/or Back of Computer/Computing Device/Peripheral Device



Optional Lip That Optional Case fits over

Numbers, letters, and/or symbols match top and bottom of board. Wires hook wherever user wants to put them. Software controls circuits.

FIG._25L

Examples of optional LCD screen dialog.

Please Wait. Repair In Progress
Please shut down computer.
Please Wait.
Please Restart Computer.

In addition to using an LCD screen, or in Lieu of using an LCD screen, dialog can be written directly to monitor, and/or communicated by other means of communication such as speech.

FIG._25M

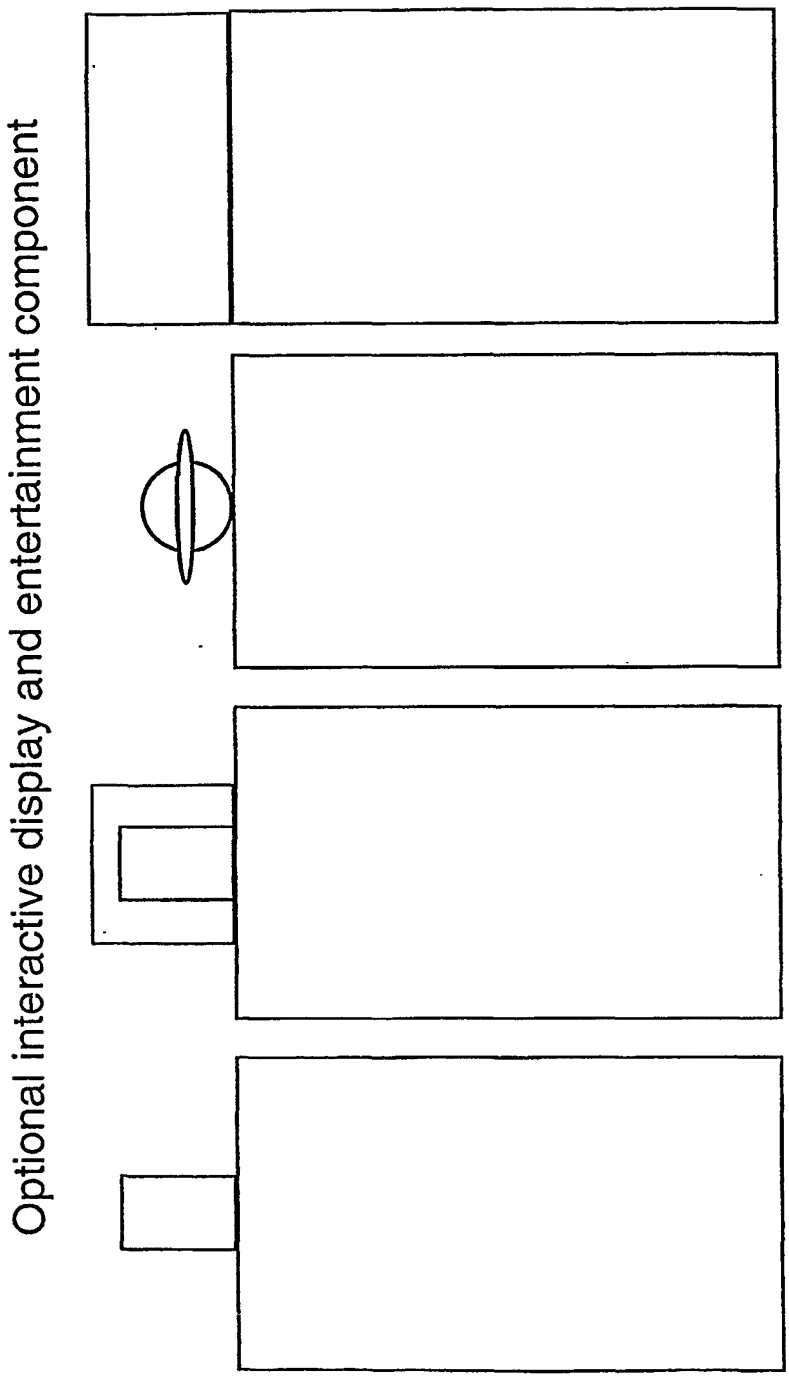

Optional interactive display and entertainment component
Optional acrylic (or similar plastic) lit by LCD to indicate the state of the DRAMUS switch. Optionally it can also be used as a decoration triggered by sound, motion, etc. Various colored lights are used to change colors in acrylic. Other art pieces can also be put in these "entertainment" boxes, such as lava lights, plasma lamps, and various art projects.

FIG._25N

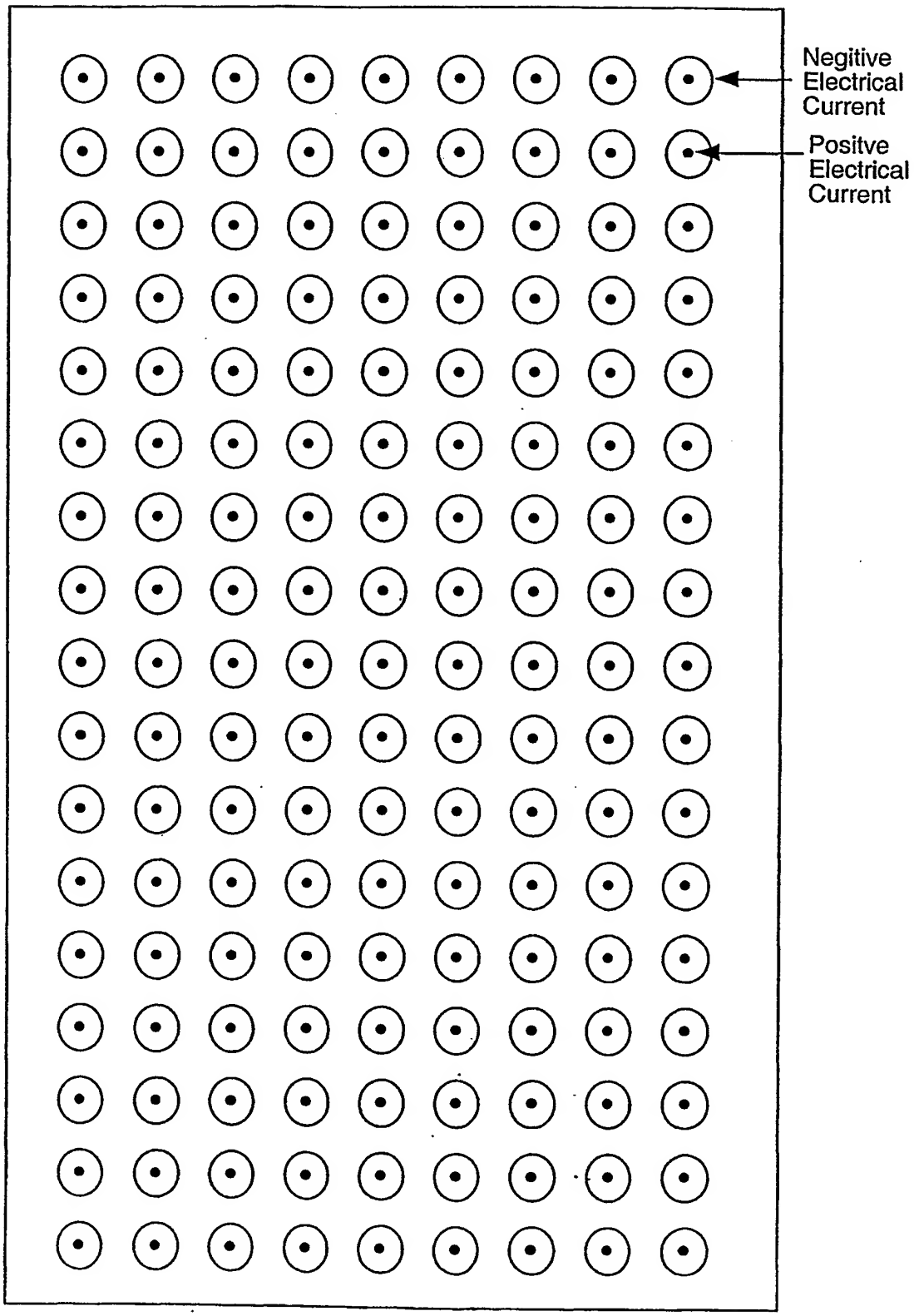
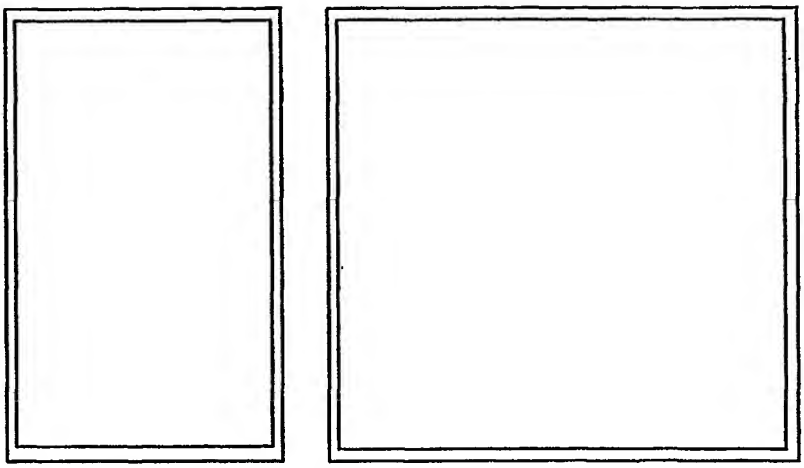
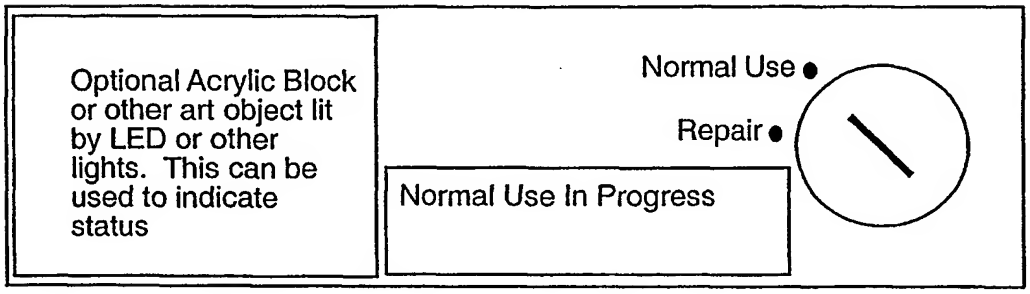


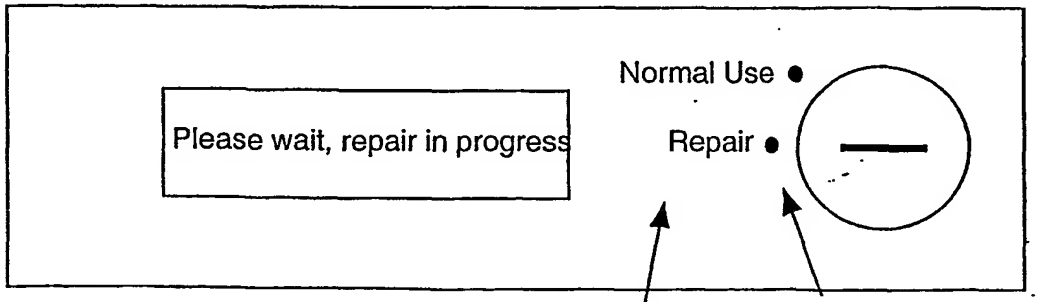
FIG._25O



Optional Colored Transparent Acrylic Case Lit by LCD or other light source. A layer of white acrylic or similar material can be under the outer layer for the purpose of diffusing the light. Optionally, case can be clear and change color when computer functions change when different groups of colored LCDs go on depending on switching functions taking place, or sounds, motion or other triggers.

FIG._25P





Optional acrylic lit by LED or other light source.

Optional LED activity lights

FIG._25Q

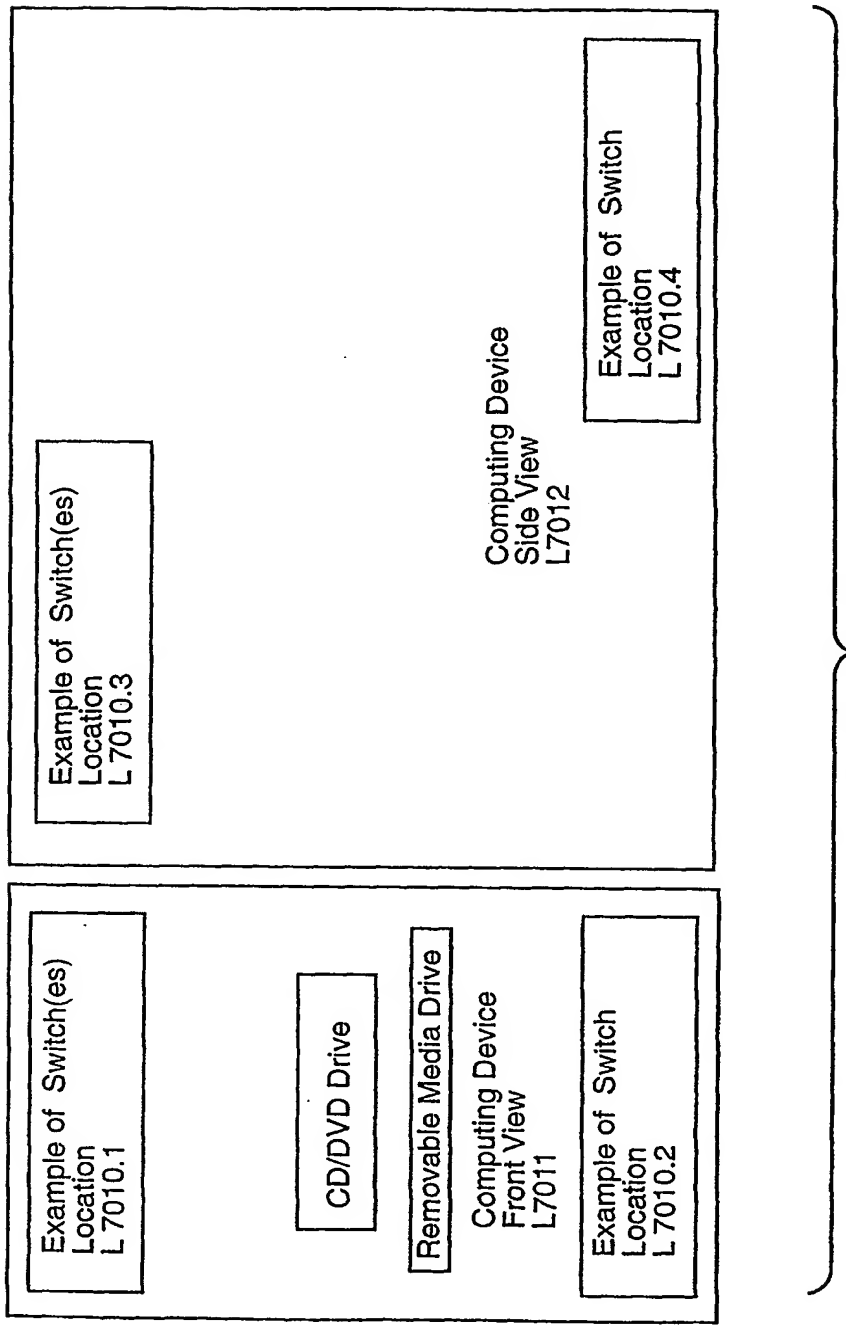

Example of Switch(es) Location L7010.1
CD/DVD Drive
Removable Media Drive
Computing Device Front View L7011
Example of Switch Location L7010.2
Example of Switch(es) Location L7010.3
Computing Device Side View L7012
Example of Switch Location L7010.4

FIG._26A

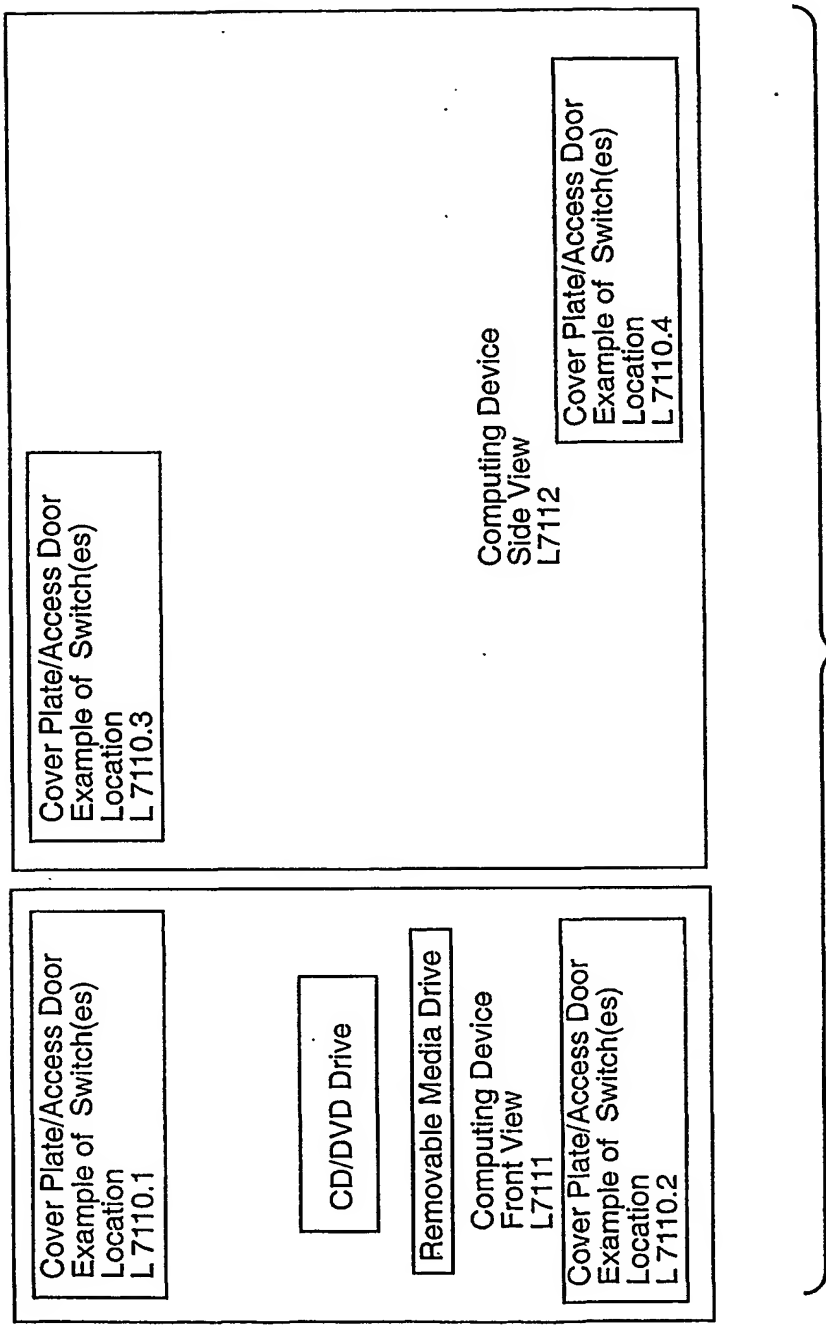

Cover Plate/Access Door Example of Switch(es) Location L7110.1
CD/DVD Drive
Removable Media Drive
Computing Device Front View L7111
Cover Plate/Access Door Example of Switch(es) Location L7110.2
Cover Plate/Access Door Example of Switch(es) Location L7110.3
Computing Device Side View L7112
Cover Plate/Access Door Example of Switch(es) Location L7110.4

FIG._26B

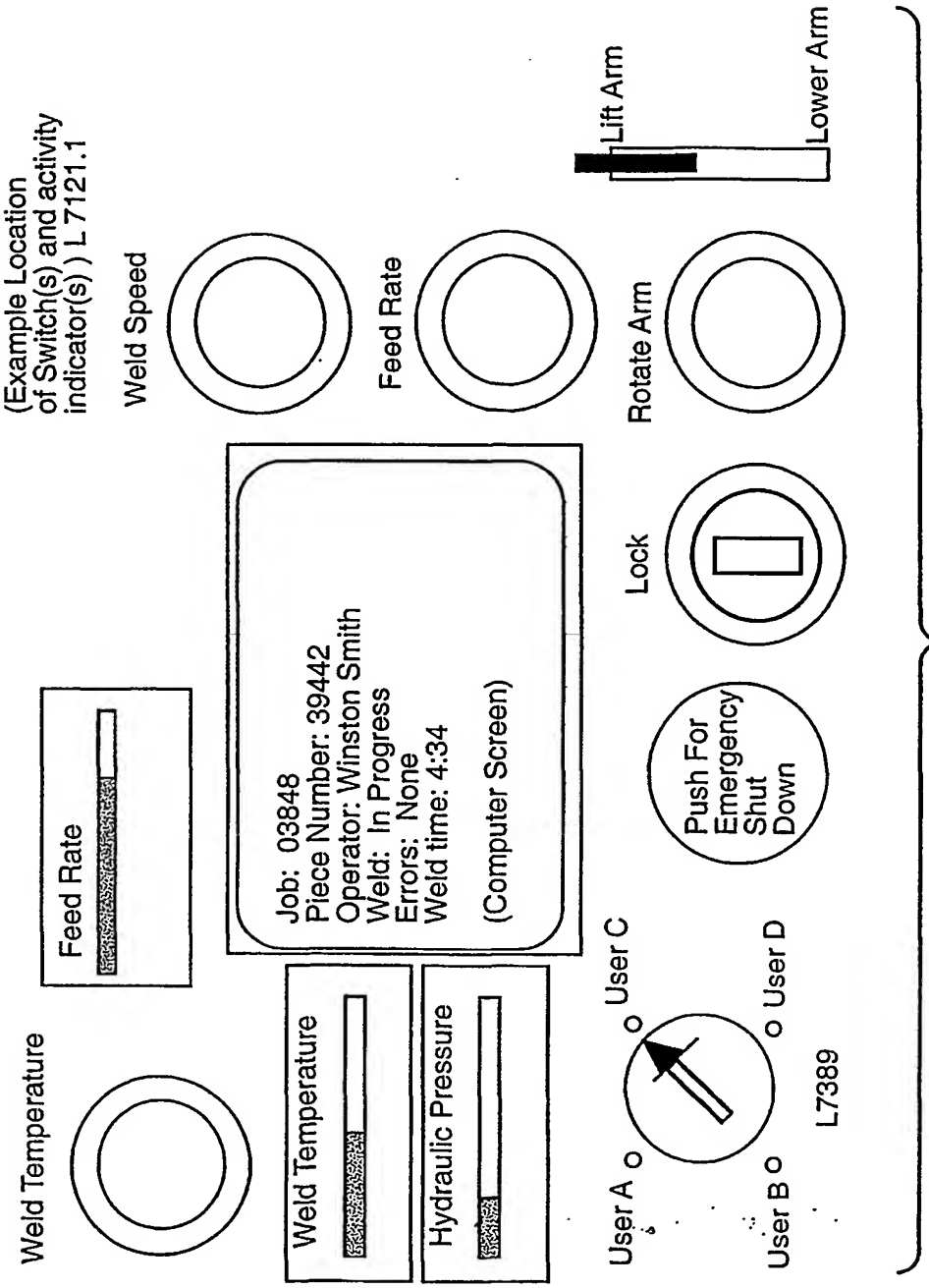

Remote SWITCH
(Example Location
of Switch(s) and activity
indicator(s)) L 7121.1
Weld Speed
Feed Rate
Rotate Arm
Lift Arm
Lower Arm
Weld Temperature
Feed Rate
Job: 03848
Piece Number: 39442
Operator: Winston Smith
Weld: In Progress
Errors: None
Weld time: 4:34
(Computer Screen)
Weld Temperature
Hydraulic Pressure
Lock
User A
User B
User C
User D
Push For
Emergency
Shut
Down
L7389

FIG._26C

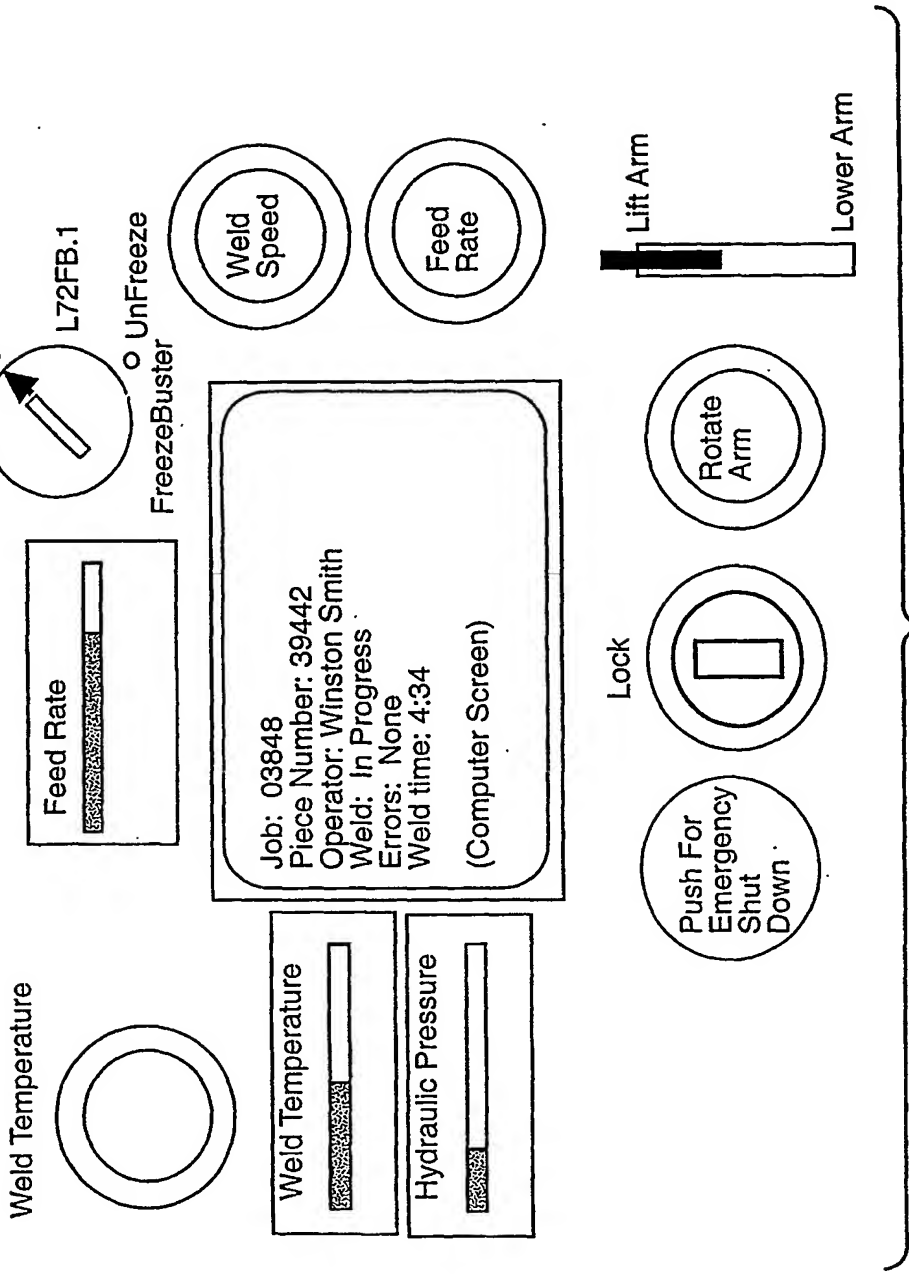

Normal Use
L72FB.1
UnFreeze
FreezeBuster
Weld Speed
Feed Rate
Weld Temperature
Feed Rate
Job: 03848
Piece Number: 39442
Operator: Winston Smith
Weld: In Progress
Errors: None
Weld time: 4:34
(Computer Screen)
Weld Temperature
Hydraulic Pressure
Lock
Lift Arm
Rotate Arm
Lower Arm
Push For Emergency Shut Down

FIG._26D

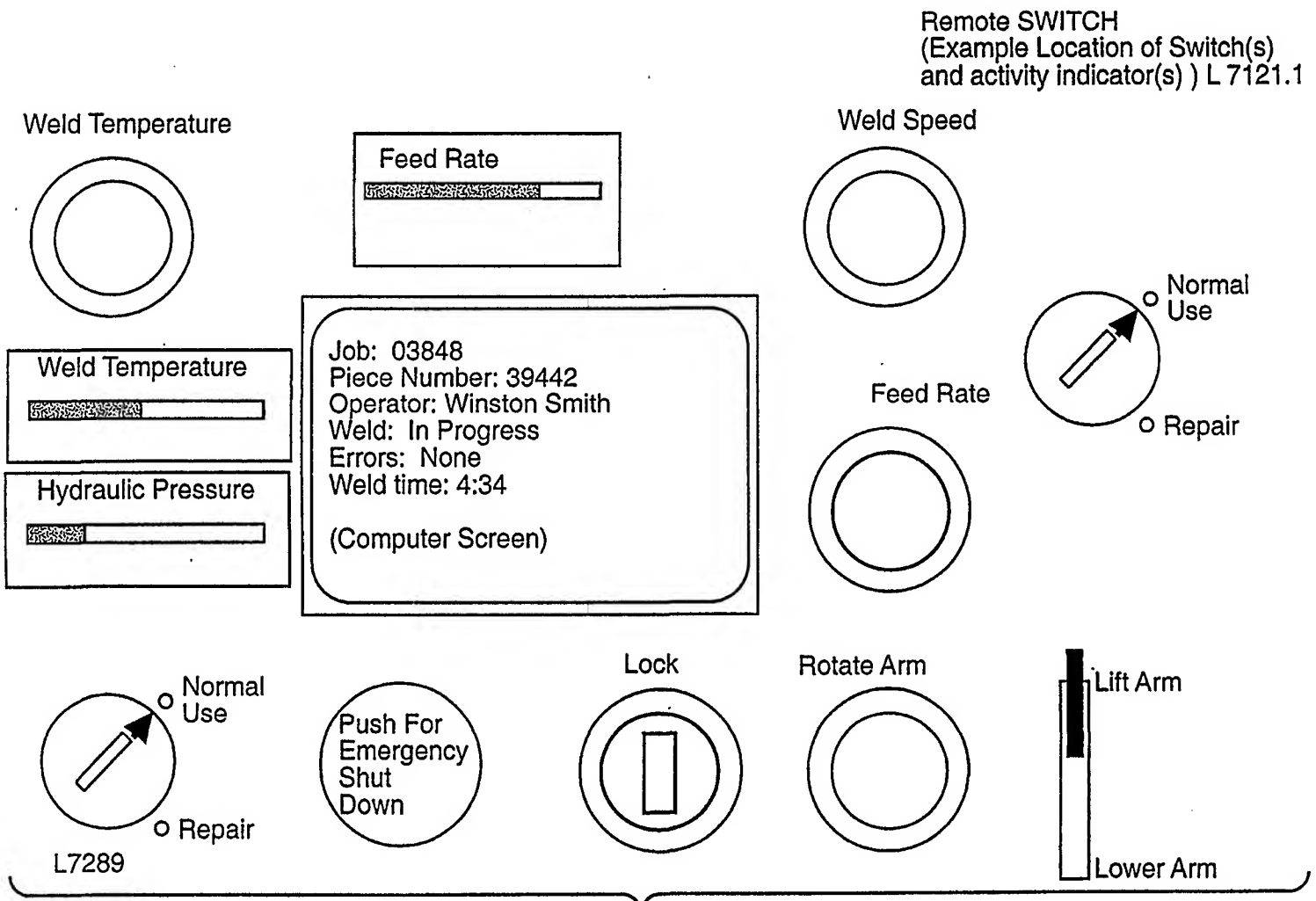

Remote SWITCH
(Example Location of Switch(s)
and activity indicator(s)) L 7121.1
Weld Temperature
Feed Rate
Weld Speed
Normal
Use
Repair
Weld Temperature
Hydraulic Pressure
Job: 03848
Piece Number: 39442
Operator: Winston Smith
Weld: In Progress
Errors: None
Weld time: 4:34
(Computer Screen)
Feed Rate
Normal
Use
Repair
L7289
Push For
Emergency
Shut
Down
Lock
Rotate Arm
Lift Arm
Lower Arm

FIG._26E

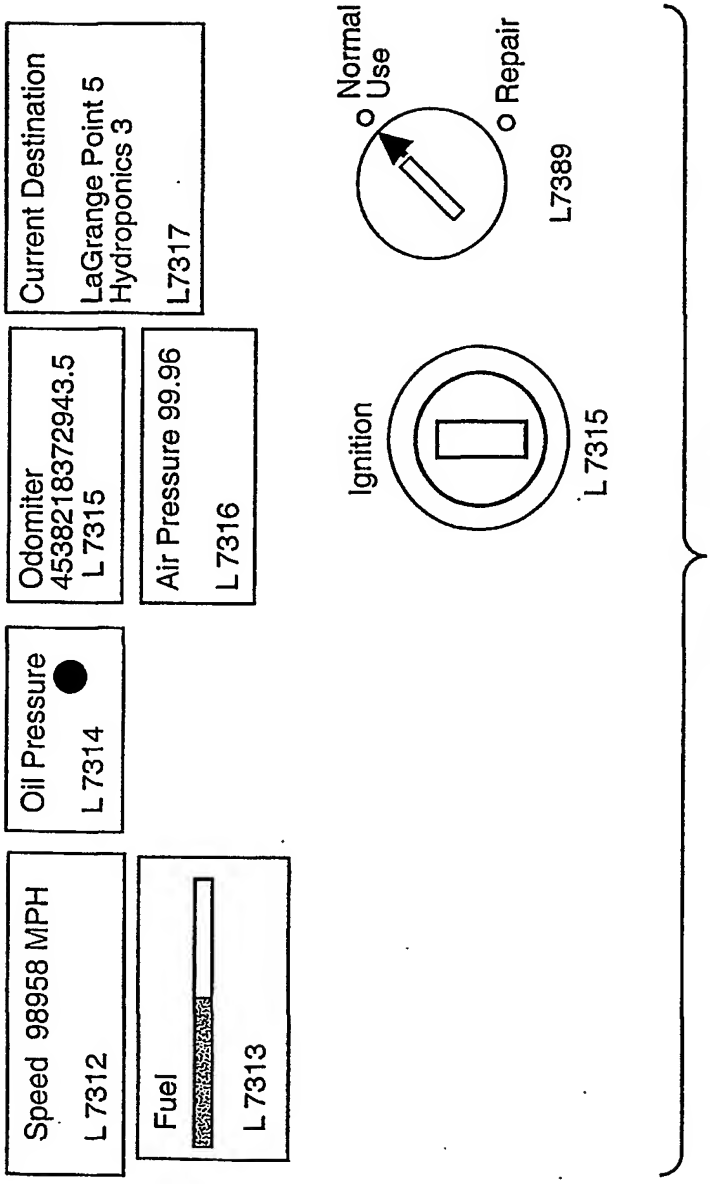

Speed 98958 MPH
L 7312
Oil Pressure
L 7314
Odomiter
4538218372943.5
L 7315
Current Destination
LaGrange Point 5
Hydroponics 3
L7317
Fuel
L 7313
Air Pressure 99.96
L 7316
Ignition
L 7315
Normal Use
Repair
L7389

FIG._26F

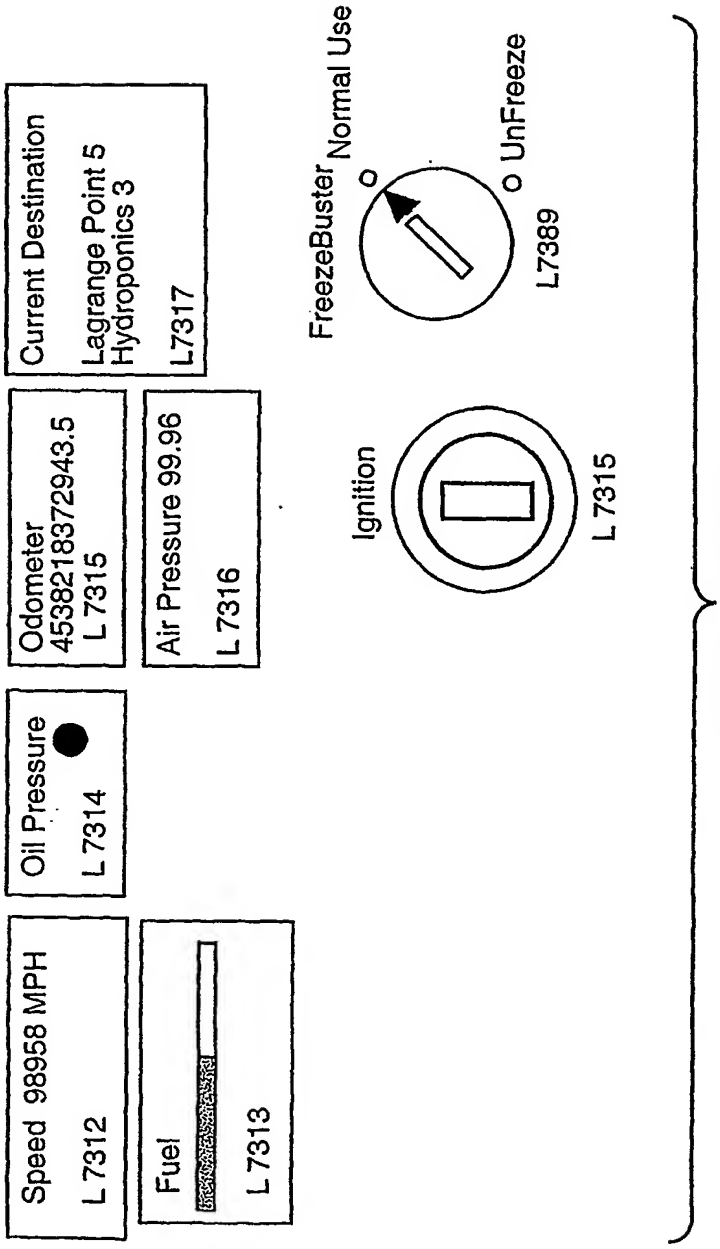

Speed 98958 MPH
L 7312
Oil Pressure
L 7314
Odometer
4538218372943.5
L 7315
Current Destination
Lagrange Point 5
Hydroponics 3
L7317
Fuel
L 7313
Air Pressure 99.96
L 7316
FreezeBuster
Normal Use
UnFreeze
L7389
Ignition
L 7315

FIG._26G

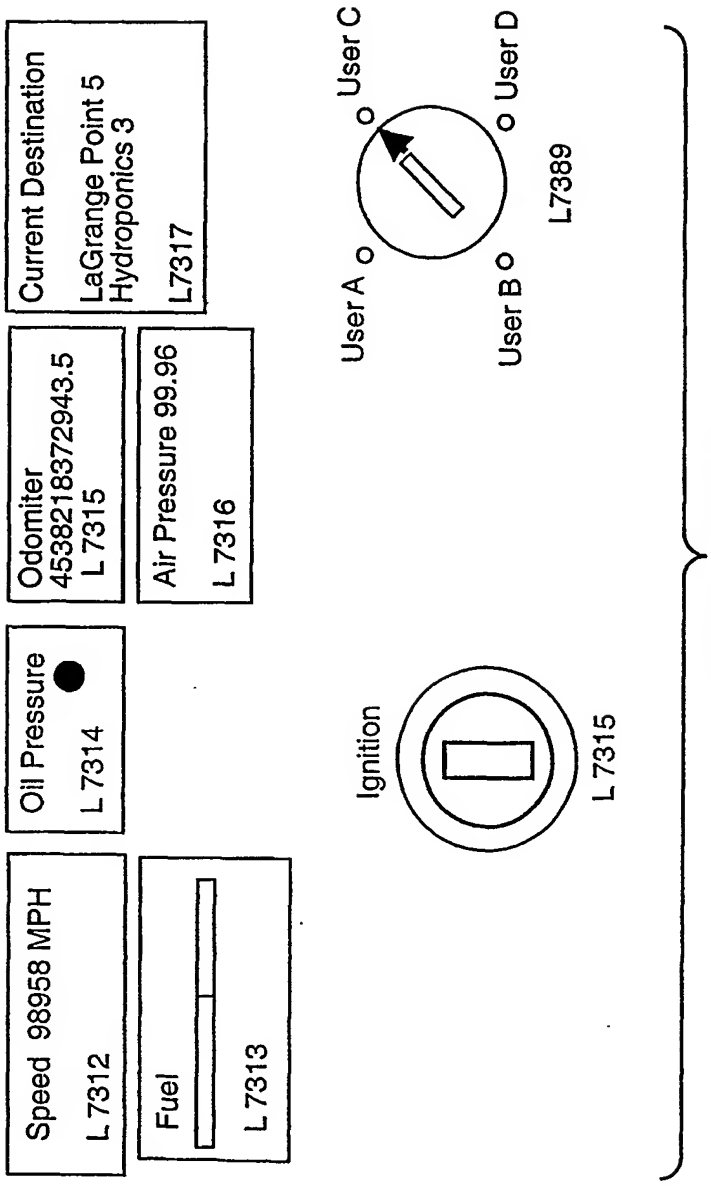

Speed 98958 MPH
L 7312
Oil Pressure
L 7314
Odomiter
4538218372943.5
L 7315
Current Destination
LaGrange Point 5
Hydroponics 3
L7317
Fuel
L 7313
Air Pressure 99.96
L 7316
User A
User B
User C
User D
L7389
Ignition
L 7315

FIG._26H

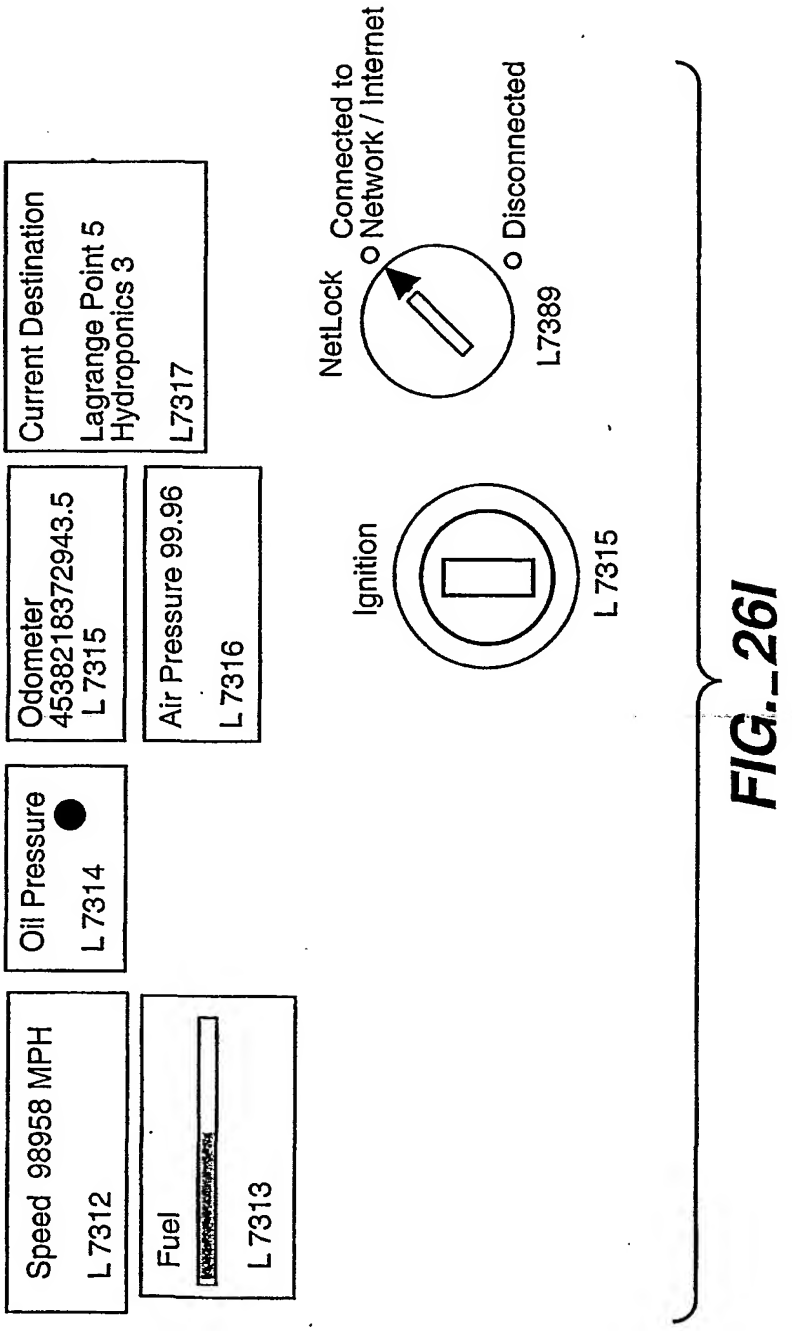

Speed 98958 MPH
L 7312
Oil Pressure
L 7314
Odometer
4538218372943.5
L 7315
Current Destination
Lagrange Point 5
Hydroponics 3
L7317
Fuel
L 7313
Air Pressure 99.96
L 7316
NetLock
Connected to
Network / Internet
Disconnected
L7389
Ignition
L 7315

FIG._26I

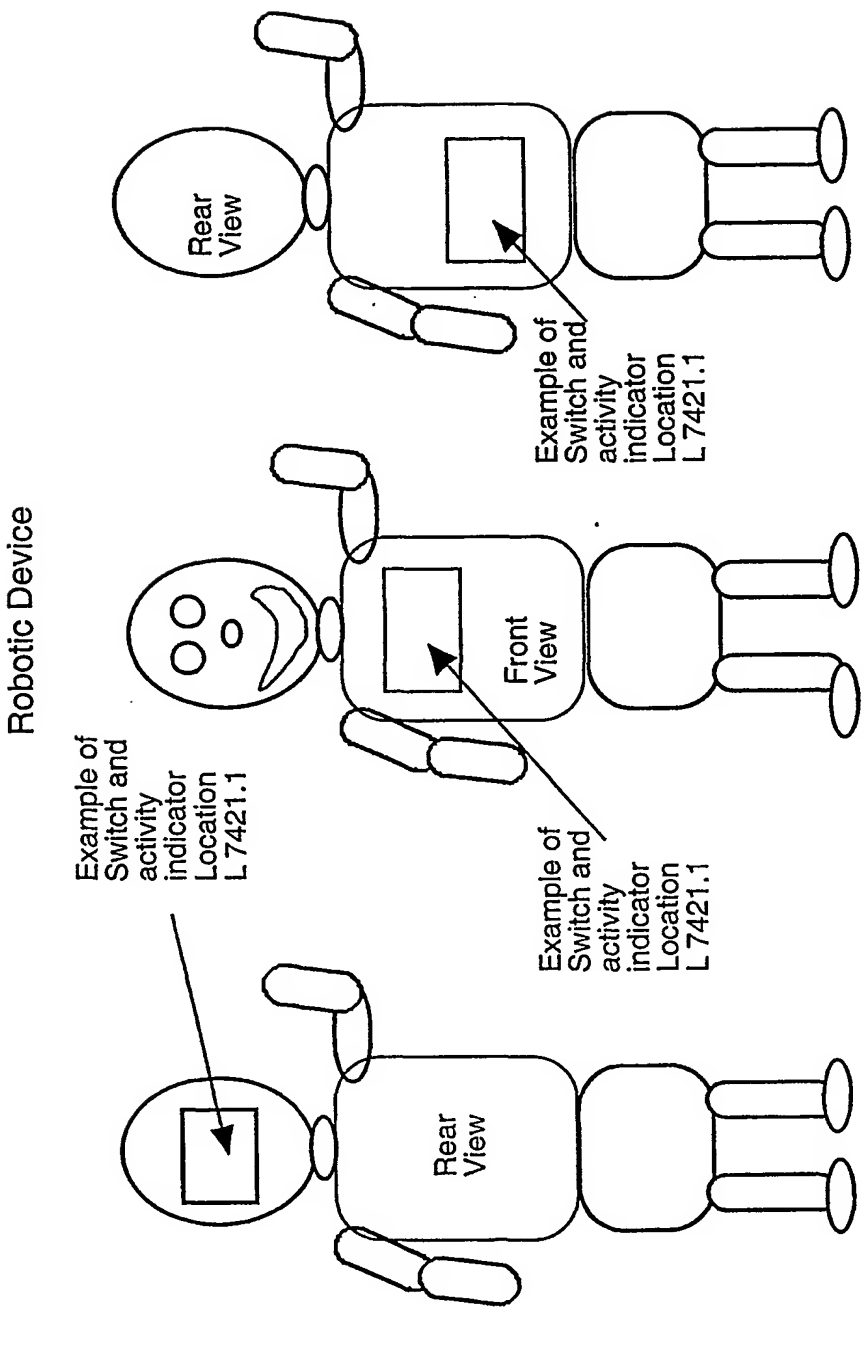

Robotic Device
Example of Switch and activity indicator Location L7421.1
Rear View
Example of Switch and activity indicator Location L7421.1
Front View
Example of Switch and activity indicator Location L7421.1
Rear View

FIG._26J

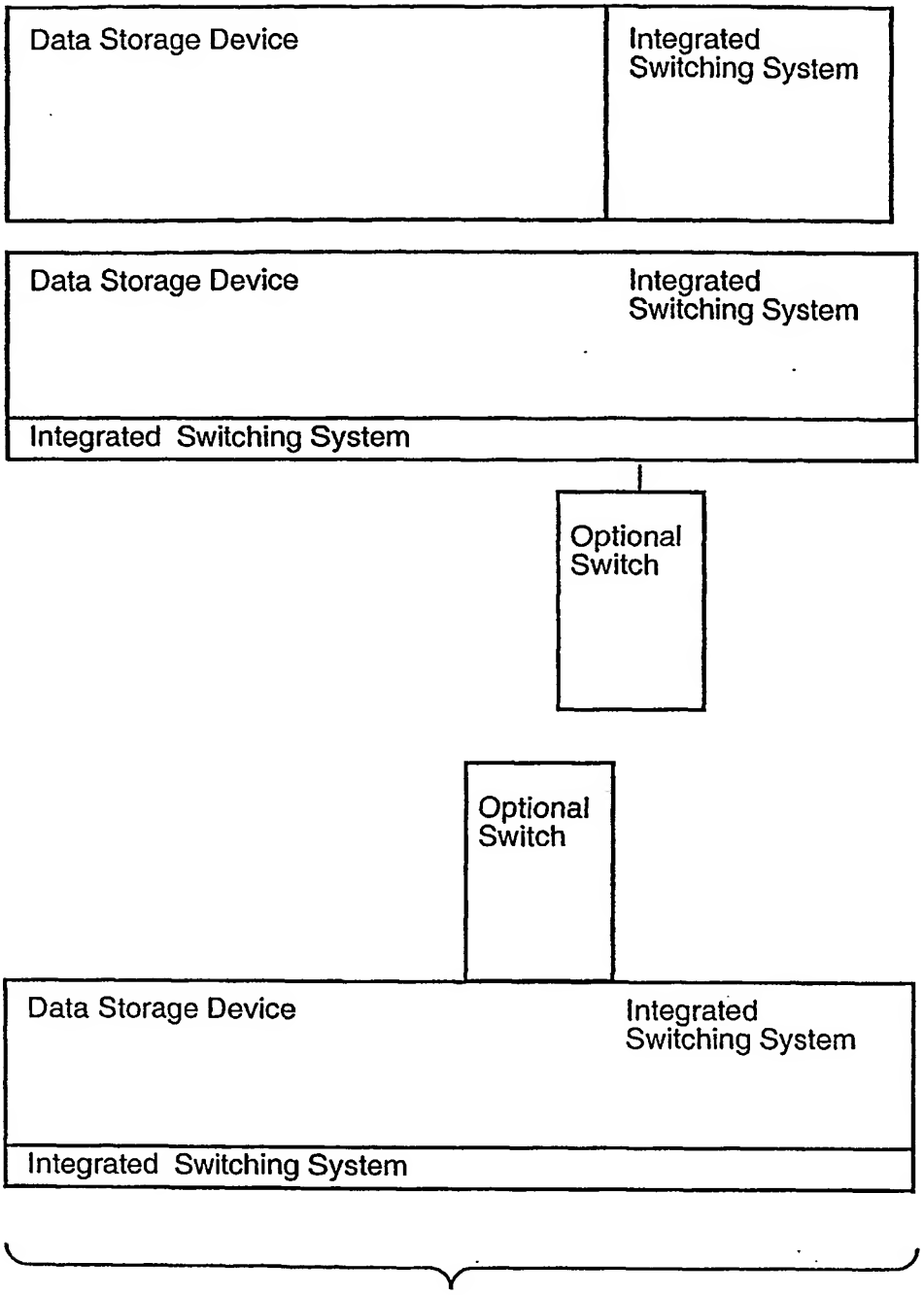

Data Storage Device
Integrated Switching System
Data Storage Device
Integrated Switching System
Integrated Switching System
Optional Switch
Optional Switch
Data Storage Device
Integrated Switching System
Integrated Switching System

FIG._26K

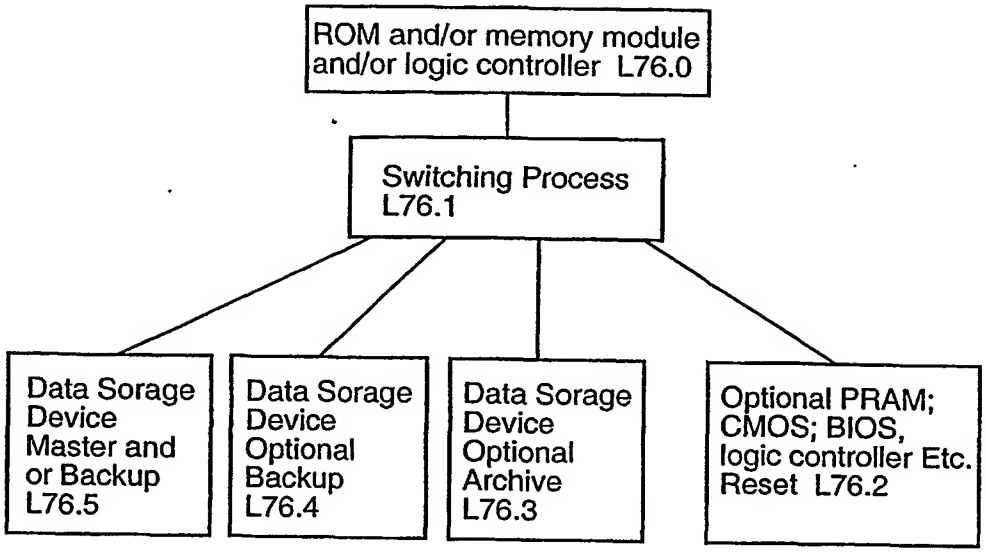


FIG._26L

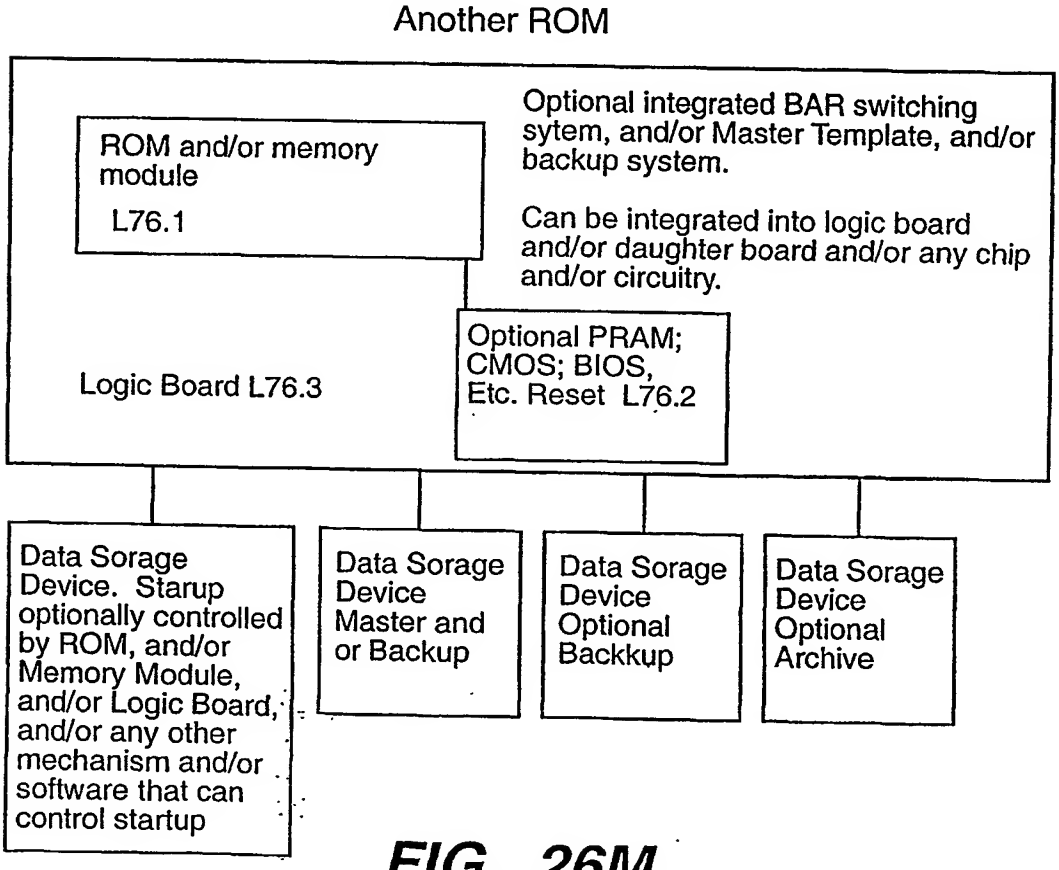


FIG._26M

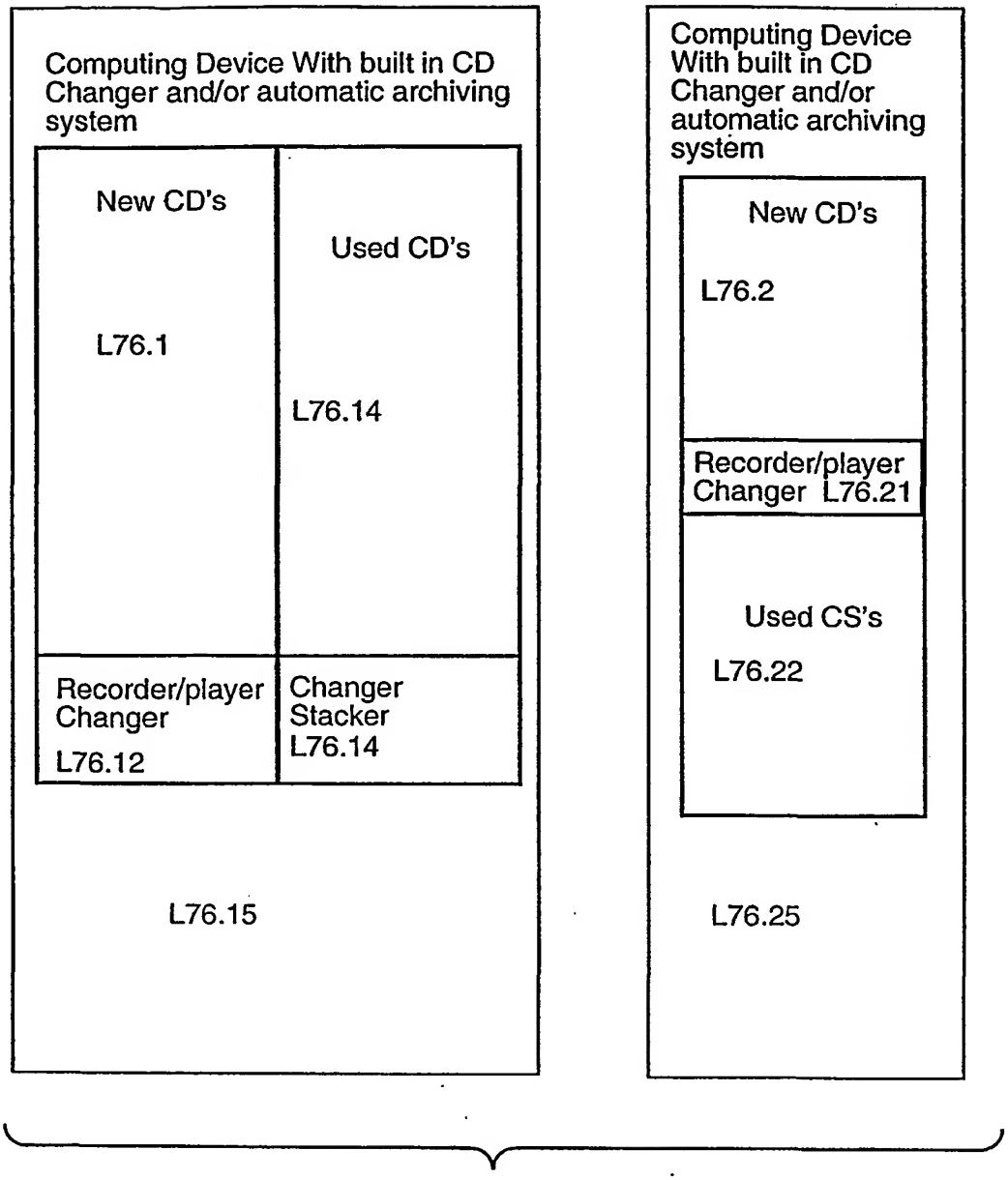

Computing Device With built in CD Changer and/or automatic archiving system
New CD's
L76.1
Used CD's
L76.14
Recorder/player Changer
L76.12
Changer Stacker
L76.14
L76.15
Computing Device With built in CD Changer and/or automatic archiving system
New CD's
L76.2
Recorder/player Changer L76.21
Used CS's
L76.22
L76.25

FIG._26N

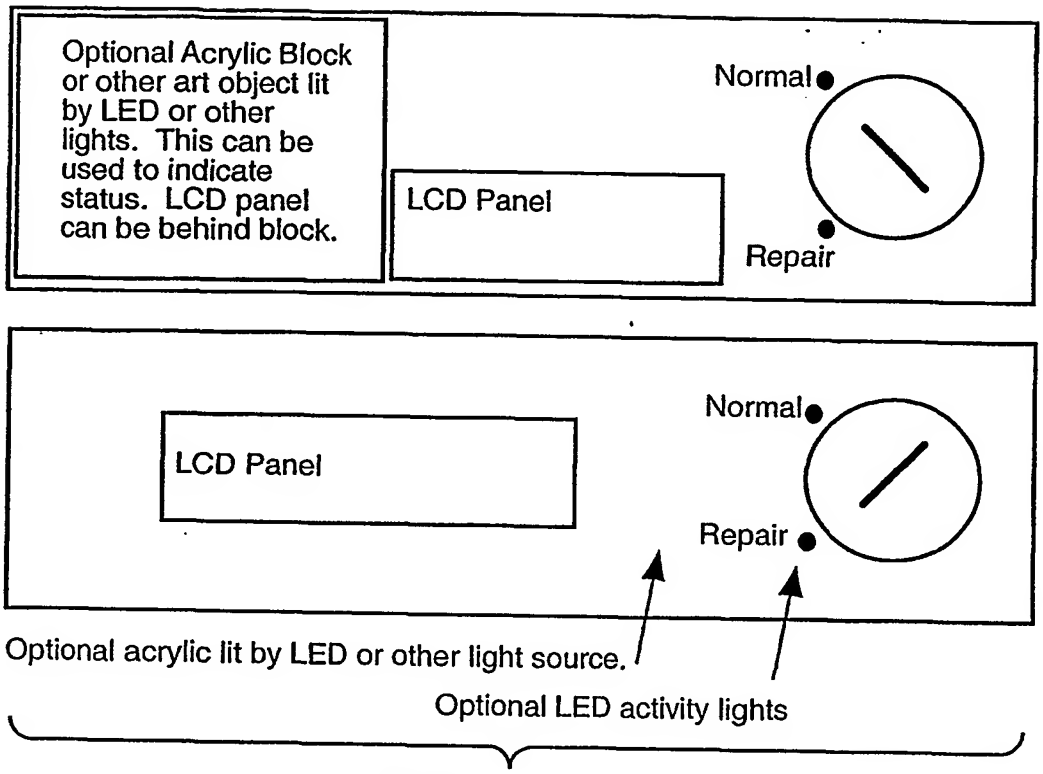


FIG._26O

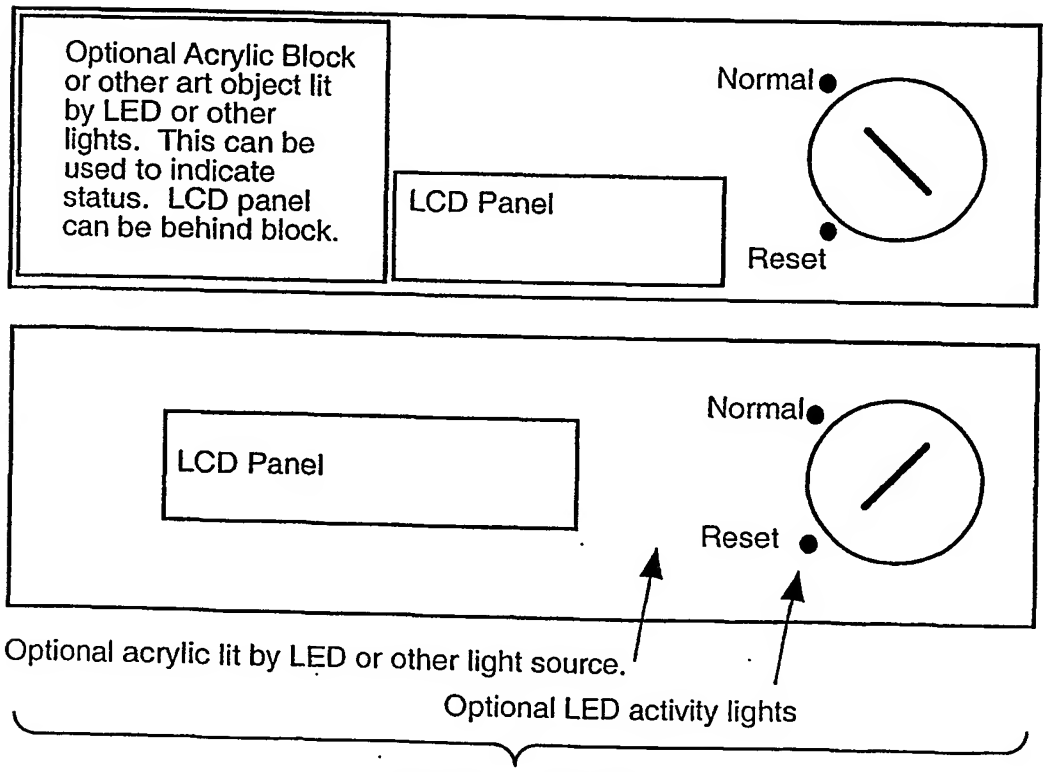


FIG._26P

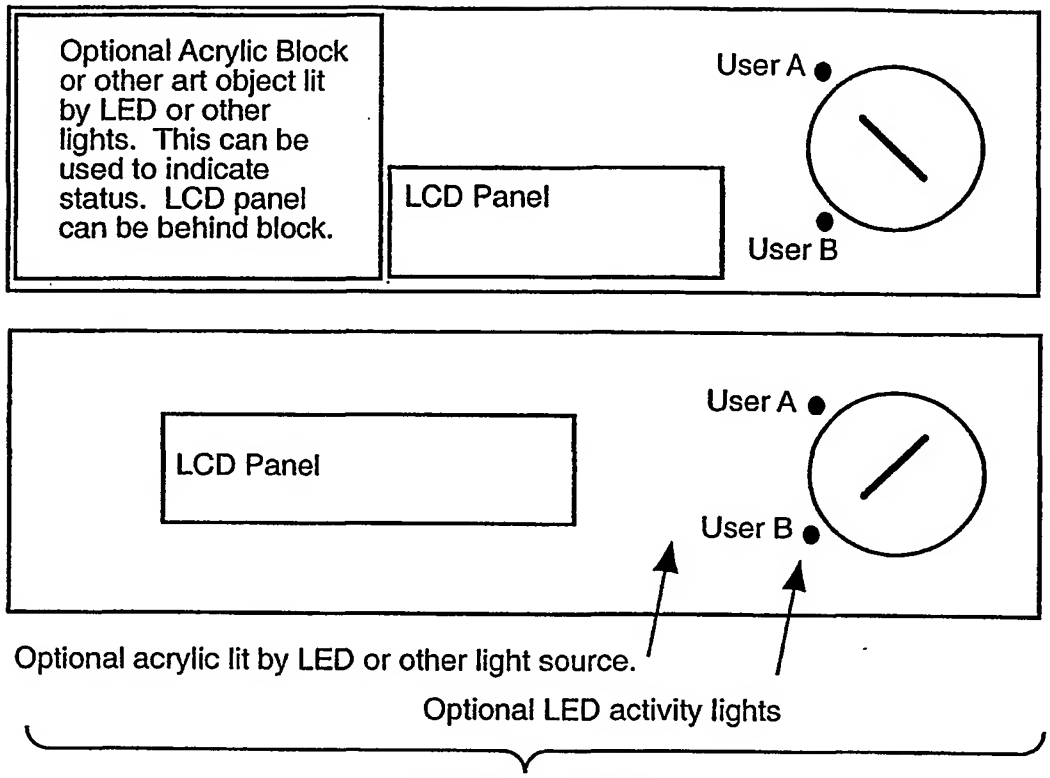


FIG._26Q

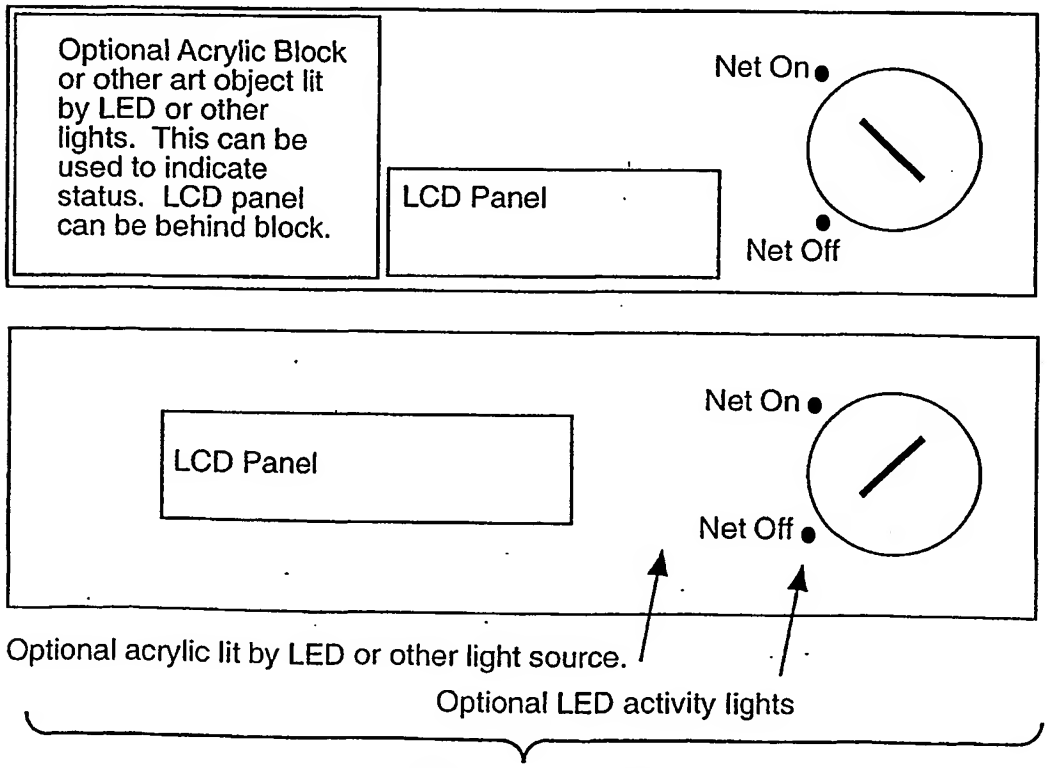


FIG._26R

Examples of optional LCD screen dialog.

Please Wait. Repair In Progress

Please shut down computer.

Please Wait.

Please Restart Computer.

In addition to using an LCD screen, or in Lieu of using an LCD screen, dialog can be written directly to monitor, and/or communicated by other means of communication such as speech.

FIG._26S

Examples of optional LCD screen dialog.

Please Wait. Switching In Progress

Please shut down computer.

Please Wait.

Please Restart Computer.

In addition to using an LCD screen, or in Lieu of using an LCD screen, dialog can be written directly to monitor, and/or communicated by other means of communication such as speech.

FIG._26T

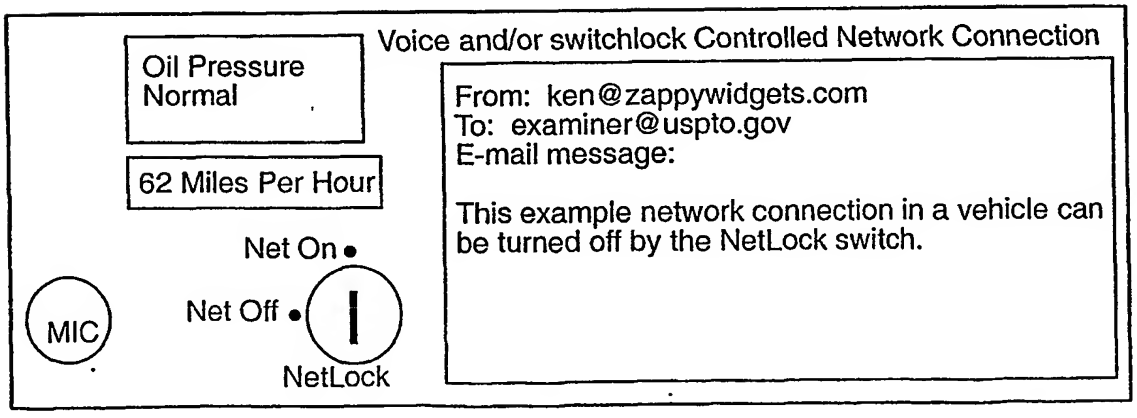
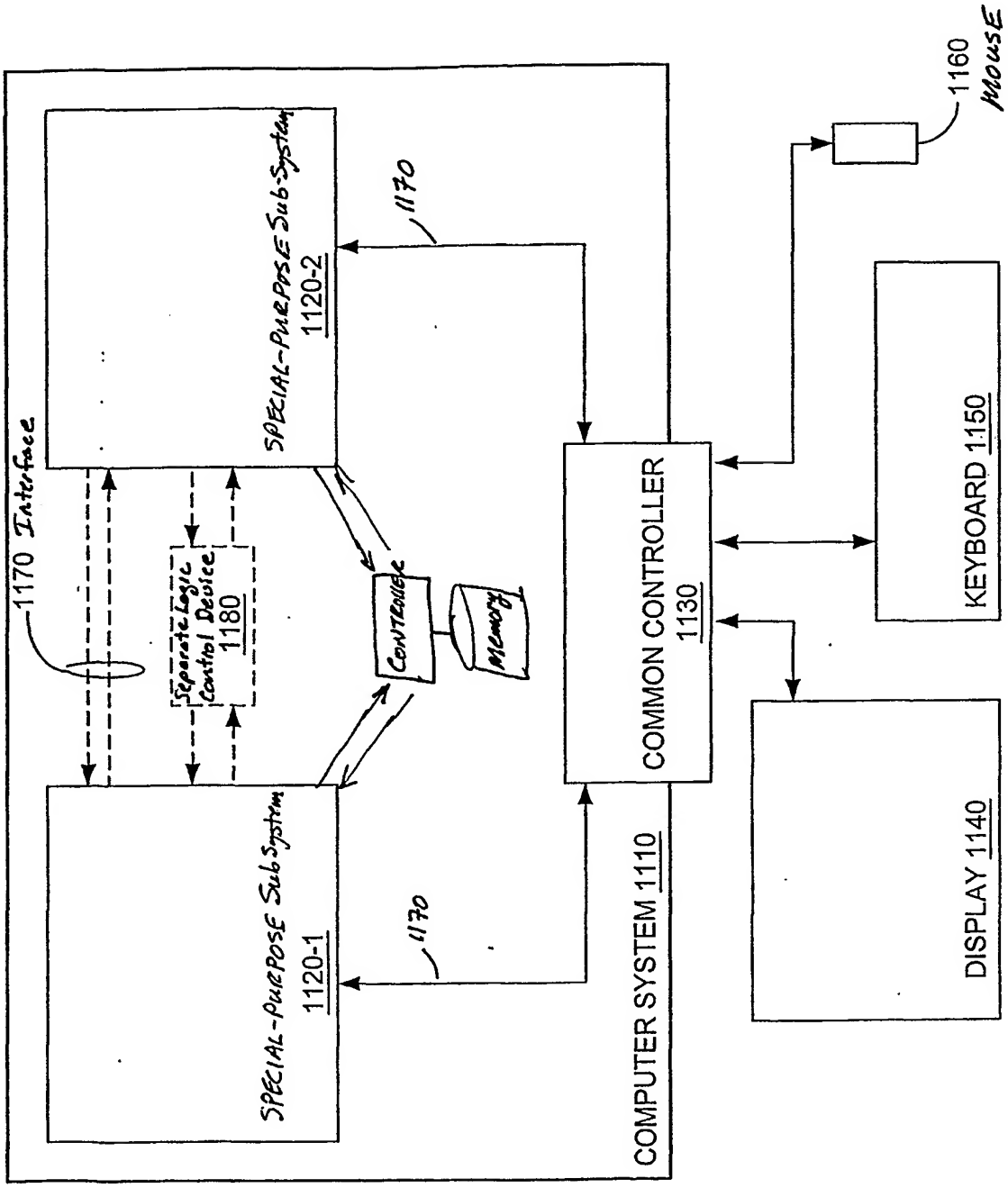
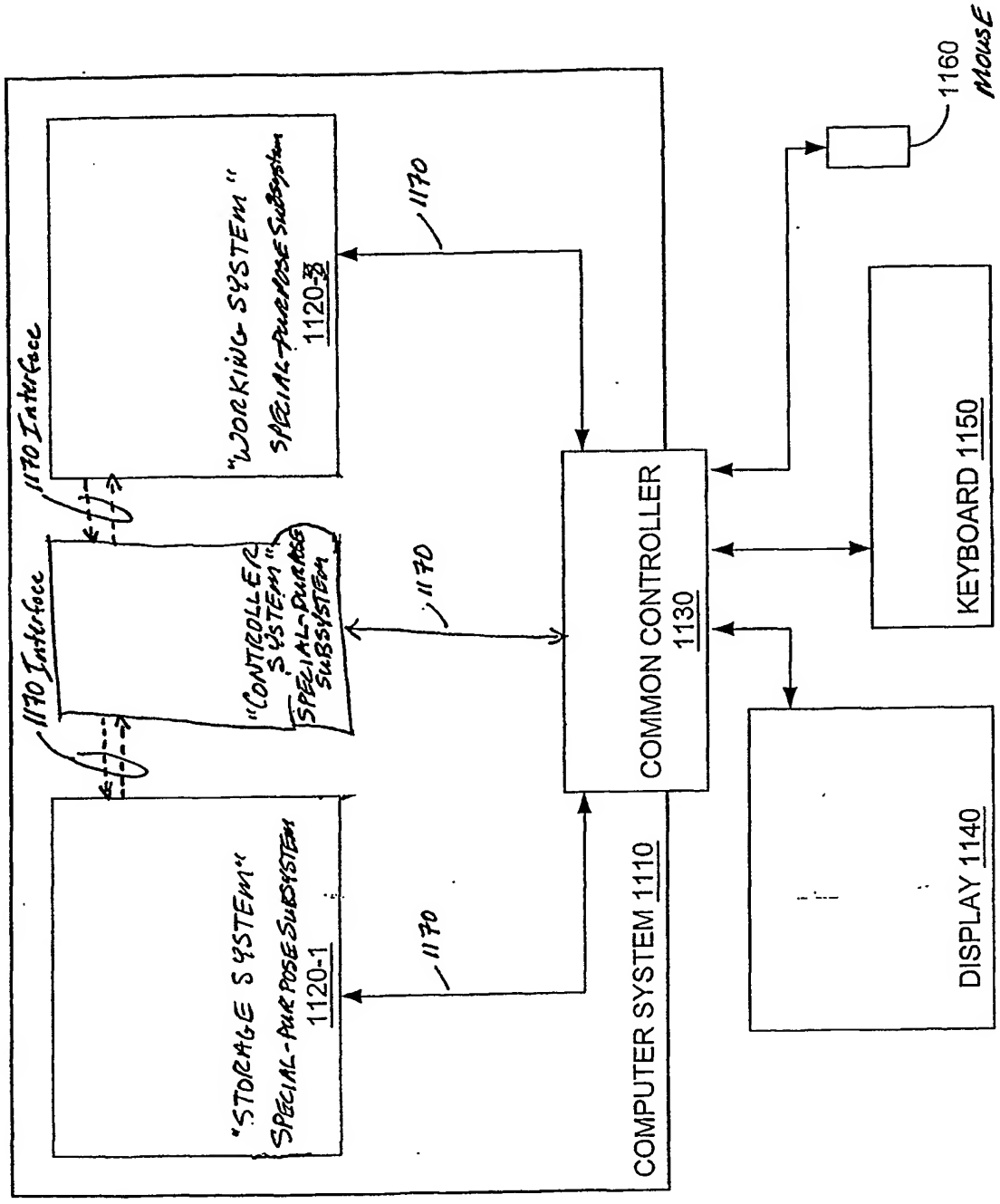


FIG._26U


1160 Mouse
1170
SPECIAL-PURPOSE Sub-System 1120-2
1170
1170 Interface
Separate Logic Control Device 1180
Controller
Memory
COMMON CONTROLLER 1130
KEYBOARD 1150
SPECIAL-PURPOSE Sub System 1120-1
1170
COMPUTER SYSTEM 1110
DISPLAY 1140


1160 Mouse
"Working System" Special-Purpose Subsystem 1120-2
1170 Interface
1170
"Controller System" Special-Purpose Subsystem
1170 Interface
1170
"Storage System" Special-Purpose Subsystem 1120-1
1170
COMMON CONTROLLER 1130
KEYBOARD 1150
DISPLAY 1140
COMPUTER SYSTEM 1110

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 02/07154

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F11/14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 978 785 A (HEWLETT PACKARD CO) 9 February 2000 (2000-02-09) abstract page 5, line 32 - line 57 page 6, line 25 - line 30 page 9, line 53 -page 10, line 7 page 11, line 10 -page 12, line 1 figure 5	1-5
X	WO 95 22794 A (APPLE COMPUTER ;YEN JOHN (US)) 24 August 1995 (1995-08-24) abstract page 7, line 21 -page 9, line 9 page 10, line 21 -page 11, line 6 page 14, line 1 - line 8 figure 2	1-5

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
1 November 2002	12/11/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Leuridan, K

1 -

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 02/07154

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0978785 A	09-02-2000	EP 0978785 A1	09-02-2000
		EP 1101164 A1	23-05-2001
		GB 2344441 A	07-06-2000
		WO 0008561 A1	17-02-2000
		JP 2002522826 T	23-07-2002
WO 9522794 A	24-08-1995	US 6381694 B1	30-04-2002
		AU 1876895 A	04-09-1995
		WO 9522794 A1	24-08-1995